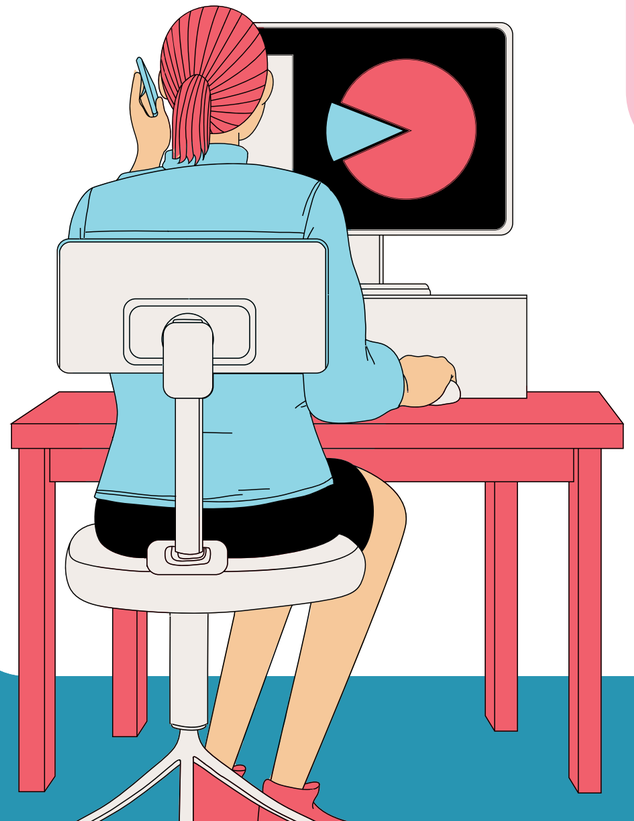
MARKET DATA FEED MONITOR



Laurie Batista - Mélanie Gonçalves -
Eliane Gardes - Clara Arnone

EXAMPLE DESCRIPTION

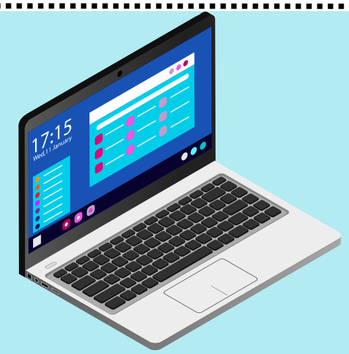
- The “Market Data Feed” processes a raw flow of market data. It detects when the data rate of a stream decreases unexpectedly.

- The “Market Data Feed” monitors market data streams in real time, calculating transaction rates per second and spot sudden drops of throughput that may indicate a problem with the data stream.

EXAMPLE SCENARIO

Finally : The simulator creates market data events for 2 feeds. It simulates drops of throughput by reducing the throughput randomly at 60% of the target rate for this second.

Detecting a drop



10 secs ago

Now

The program detects a drop of throughput by comparing the actual number of ticks per second with an average calculated on the last 10 seconds. If the actual throughput drops below 75%, it reports a possible drop

Calculation of rates by flow

TicksPerSecond

Events that come from MarketDataEvent are sorted by second in TicksPerSecond.The program calculates the number of transactions (ticks) per second for each market data feed. It groups events by stream and calculates the number of events per second (ticks per second) using a one second time window.

Entry event

Raw data

MarketDataEvent

Events representing market data (security symbol, buy price and sell price) arrive in mass in the Market Data Event

TicksPerSecondStatement

EPL QUERY DETAILS

```
    public TicksPerSecondStatement(EPDeploymentService admin, Configuration configuration) {
        String stmt = "insert into TicksPerSecond " +
            "select feed, count(*) as cnt from MarketDataEvent#time_batch(1 sec) group by feed";
        CompilerArguments args = new CompilerArguments(configuration);
        args.getOptions().setAccessModifierEventType(env -> NameAccessModifier.PUBLIC); // export the
type allowing others to refer to it
        try {
            EPCompiled compiled = EPCompilerProvider.getCompiler().compile(stmt, args);
            statement = admin.deploy(compiled).getStatements()[0];
        } catch (Exception ex) {
            throw new RuntimeException(ex);
        }
    }

    public void addListener(UpdateListener listener) {
        statement.addListener(listener);
    }
```

“select feed, count (*) as cnt”

Cette requête selectionne la requête “feed” et l’occurence de chaque valeur dans “feed” renommée “cnt”

“from MarketDataEvent#time_batch (1 sec)”

Elle précise la source des événements d’où les données seront extraites et spécifie la fenêtre temporelle “**time_batch (1 sec)**”.

“group by feed”

Elle précise que les résultats doivent être regroupés par la colonne feed

TicksFalloffStatement

EPL QUERY DETAILS

```
public class TicksFalloffStatement {
    private EPStatement statement;

    public TicksFalloffStatement(EPDeploymentService admin, Configuration configuration,
EPCompilerPathable epRuntimePath) {
        String stmt = "select feed, avg(cnt) as avgCnt, cnt as feedCnt from TicksPerSecond#time(10
sec) " +
            "group by feed " +
            "having cnt < avg(cnt) * 0.75 ";

        CompilerArguments args = new CompilerArguments(configuration);
        args.getPath().add(epRuntimePath);
```

“select feed, avg (cnt) as avgCnt, cnt as feedCnt from TicksPerSecond#time (10 sec)”

Cette requête sélectionne la colonne “feed”, la moyenne des valeurs de “cnt”, ces valeurs sont sélectionnées dans la fenêtre temporelle de 10 sec de l’event “TicksPerSecond”

“group by feed”

Elle précise que les résultats doivent être regroupés par la colonne feed

“having cnt < avg (cnt) * 0,75 ”

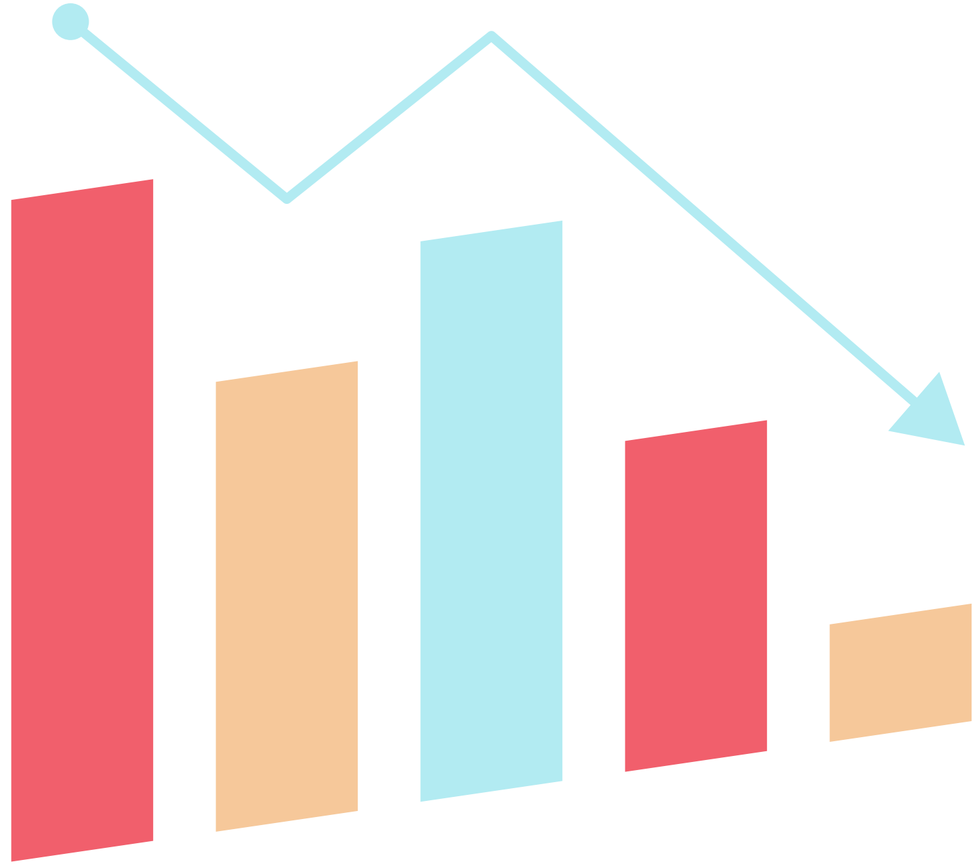
Elle filtre les résultats en gardant les lignes où cnt est inférieur à 75% de l’avg (cnt) pour chaque groupe de “feed”

EPL RESULTS

Le résultat de la requête nous montre donc le simulateur nous a créé des événements de données pour 2 flux : FEED A et FEED B.
Il va simuler des baisses de débit en réduisant de 75% du débit cible aléatoirement : “setting drop-off for feed FEED A”.

Le taux va baisser pour FEED A.

Le simulateur alors detecter la baisse pour le FEED A, nous donner son taux, et nous donner la moyenne des feeds dans le thread



THANK YOU :)