Tracking a Soccer Game with Big Data

Asanka Abeysinghe
Vice President, Solutions Architecture - WSO2, Inc

QCon Sao Paulo - 2015
Story about soccer
and Big Data
Outline

- Big Data and CEP
- Tracking a Soccer Game
- Making it real
- Conclusion

Photo credit John Trainor on Flickr http://www.flickr.com/photos/trainor/2902023575/, Licensed under CC
Think about a day in your life?
- What is the best road to take?
- Would there be any bad weather?
- How to invest my money?
- How is my health?

You can make better decisions if only you can access data and process them.
Real-time analytics

- Most of the first use-cases were of batch processing, which take minutes if not hours to run.
- Many insights you’d rather have sooner:
  - Tomorrow’s weather
  - Heart condition
  - Traffic congestion
  - Natural disaster
  - Stockmarket Crash
Realizing real-time analytics

- Processing Data on the fly, while storing a minimal amount of information and responding fast (from <1 ms to few seconds)

- Idea of Event streams
  - A series of events in time

- Enabling technologies
  - Stream Processing (Storm, S4)
  - Complex Event Processing
Internet of Things (IoT)

- Currently the physical world and software worlds are detached
- Internet of things promises to bridge this
  - It is about sensors and actuators everywhere
  - In your fridge, in your blanket, in your chair, in your carpet.. Yes even in your socks
  - Google IO pressure mats
Vision of the future

- Sensors everywhere
- Data collected from everywhere, analyzing, optimizing, and helping (and hopefully not taking over)
- Analytics and Internet of things. Immersive world
- Big data and real-time analytics will be crucial. How far are we from realizing that?
What required to build such world

- Sensors and actuators (Motes?)
- Fast interoperable event systems (MQTT?)
- Powerful query languages (CEP?)
- Powerful control systems and decision systems
Data processing landscape
Data processing landscape

Size of the Data Handled (per second)
- 100k events (100MBs)
- 1k events (1MBs)
- 100 events (10 KBs)

In-Memory Computing (Spark, Hana, VoltDB)
Interactive Processing (e.g. Drill, BigQuery, OLAP)
Indexed Storage (RDBMS, Solar)

Realtime Analytics (CEP, Stream Processing)

MapReduce (e.g. Spark, Hadoop)

Time to Act
- millis
- seconds
- minutes
- hours
- days
Complex Event Processing

from stockQuotes#window.time(1 min)
join Tweets#window.time(1 min)
on StockQuotes.symbol==Tweets.company
select *
insert into PredictedstockQuotes;

** Performance
1.8M Events/Sec on same JVM &
About 250K over Network

SOAP / REST with JSON/XML/Text
SMS/E-mail
wso2 Events (Thrift)
Other Systems

Filter Transformation Window +
{Aggregation, group by}
Join Event Sequence Event Table
Complex Event Processing

from stockQuotes#window.time(1 min)
join Tweets#window.time(1 min)
on StockQuotes.symbol==Tweets.company
select *
insert into PredictedstockQuotes;

Filter Transformation Window +
{Aggregation, group by}
Join Event Sequence Event Table

WSO2 Complex Event Processor

Stream 1
Stream Y
Stream X
Stream 1
Stream 1

**Performance**
1.8M Events/Sec on same JVM &
About 250K over Network

SOAP / REST with
JSON/XML/Text

SMS/E-mail

wso2 Events (Thrift)

Other Systems

© WSO2
CEP = SQL for real-time analytics

- Easy to follow from SQL
- Expressive, short, and sweet.
- Define core operations that covers 90% of problems
- Lets experts dig in when they like!
CEP operators

- Filters or transformations (process a single event)
  
  ```
  from Ball[v>10] select .. insert into ..
  ```

- Windows + aggregation (track window of events: time, length)
  
  ```
  from Ball#window.time(30s) select avg(v) ..
  ```

- Joins (join two event streams to one)
  
  ```
  from Ball#window.time(30s) as b join Players as p on p.v < b.v
  ```

- Patterns (state machine implementation)
  
  ```
  from Ball[v>10], Ball[v<10]*, Ball[v>10] select ..
  ```

- Event tables (map a database as an event stream)
  
  ```
  Define table HitV (v double) using .. db info..
  ```
Soccer use-cases

- Dashboard on game status
- Alarms about critical events in the game
- Real-time game analysis and predictions about the next move
- Updates/ stats etc., on mobile phone with customized offers
- Study of game and players effectiveness
- Monitor players health and body functions
DEBS challenge

- Soccer game, players and ball has sensors $\text{sid, ts, x,y,z, v,a}$
- Use cases: Running analysis, Ball Possession and Shots on Goal, Heatmap of Activity
- WSO2 CEP (Siddhi) did 100K+ throughput
Use-case 1: running analysis

Detect when speed thresholds have passed

Define partition player by Players .id;

From s = Players [v <= 1 or v > 11],
    t = Players [v > 1 and v <= 11],
    e = Players [v <= 1 or v > 11]

Select s.ts as tsStart, e.ts as tsStop, s.id as playerId,
    "trot" as intensity, t [0].v as instantSpeed,
    (e.ts - s.ts )/1000000000 as unitPeriod

Insert into RunningStats partition by player;
Use-case 2: ball possession

You possess the ball from time you hit it until someone else hit it or ball leaves the ground.

Partition By Player

Ball

Players

Join when distance < 1m and a > 55m/s2

Detect when Ball Possession Changes

from old = hitStream,
\[ b = \text{hitStream}[\text{old.pid} \neq \text{pid}], \quad n = \text{hitStream}[\text{b.pid} = \text{pid}]^*, \]
\[ e1 = \text{hitStream}[\text{b.pid} \neq \text{pid}] \text{ or } e2 = \text{ballLeavingHitStream} \]
Use-case 3: heatmap of activity

Show where actions happened (via cells defined by a grid of 64X100 etc.), need updates once every second.

Can resolve via cell change boundaries, but does not work if one player stays more than 1 sec in the same cell. Therefore, need to join with a timer.
Use-case 4: detect kick on the goal

Detect kicks on the ball, calculate direction after 1m, and keep giving updates as long as it is in right direction.
Results from scenarios

Throughput by Scenario

Throughput (Events per Second)

- Use Case 1
- Use Case 2
- Use Case 3
- Use Case 4
- All

Throughput values: Use Case 2 > Use Case 4 > Use Case 1 > Use Case 3 > All
Demo
Making it real
CEP high-availability

Option 1: Side by Side

Load Balancer (e.g. HAProxy or HW Load balancer)

CEP (Active)

Option 2: Snapshot and Restore

Load Balancer (e.g. HAProxy or HW Load balancer)

CEP (Passive)

Notification Receivers
Compare CEP vs SP

Complex Event Processing

- SQL like language
- Supports powerful temporal operators (e.g. windows, event patterns)
- Focus on speed
- Harder to scale
- e.g. WSO2 CEP, Streambase, Esper

Stream Processing

- Operators connected in a network, but you have to write the logic
- Distributed by design
- Focus on reliability (do not lose messages), has transactions
- e.g. Storm, S4
Scaling CEP: pipeline

from StockQuote[ symbol == GOOG and prize > 1500 ] insert into GOOGStocks from q1=GOOGStocks-> GOOGStocks[(value - q1.value)/q.value > 0.3] within 30s
Scaling CEP: operators

```
from Ball#window(100ms) as b
join Player as p on toM(b.loc) == toM(p.loc)
```
Siddhi Storm bolt

We have written a Siddhi bolt that would let users run distributed Siddhi Queries using Storm

```java
SiddhiBolt siddhiBolt1 = new SiddhiBolt( .. siddhi queries .. );
SiddhiBolt siddhiBolt2 = new SiddhiBolt( .. siddhi queries .. );
TopologyBuilder builder = new TopologyBuilder();
builder.setSpout("source", new PlayStream(), 1);
builder.setBolt("node1", siddhiBolt1, 1)
    .shuffleGrouping("source", "PlayStream1");
...
builder.setBolt("LeafEcho", new EchoBolt(), 1)
    .shuffleGrouping("node1", "LongAdvanceStream");
...
cluster.submitTopology("word-count", conf, builder.createTopology());
```
Siddhi Storm bolt

from StockQuotes#window.time(1 min)
join Tweets#Window.time(1 min)
on StockQuotes.symbol == Tweets.company
select *
insert into PredictedStockquotes;
Agent agent = new Agent(agentConfiguration);
publisher = new AsyncDataPublisher("tcp://localhost:7612", .. );

StreamDefinition definition =
    new StreamDefinition(STREAM_NAME, VERSION);
definition.addPayloadData("sid", STRING);
...
publisher.addStreamDefinition(definition);
...
Event event = new Event();
event.setPayloadData(eventData);
publisher.publish(STREAM_NAME, VERSION, event);

- Can receive events via SOAP, HTTP, JMS, ..
- WSO2 Events is blazing fast (400K events TPS)
- Default Agents and you can write custom agents.
Business Activity Monitor (BAM)
Think CEP AND Hive, not OR

- Real power comes when you can join the batch (NRT) and real-time processing
- E.g. if velocity of the ball after a kick is different from season average by 3 times of season’s standard deviation, trigger event with player ID and speed
  - Hard to detect velocity after the kick with Hive/MapReduce, but much easier with CEP
  - Complicated to calculate season average with CEP, as we end up up with huge windows.
CREATE EXTERNAL TABLE IF NOT EXISTS AvgSpeed ... select avg(v), std(v) FROM SPEEDS

(player, time, v)

{Hive : Calculate Season’s Average}

{CEP: Detect kicks}

{CEP: Join and match against Season’s Average}

FROM AfterKickVStream
    #window.length(0) AS k
JOIN AvgSpeed AS a
ON k.v > v + 3*std
INSERT INTO FastHits;
Other real-world use-cases

- System/ Device Management
- Fleet/ Logistic Management
- Fraud Detection
- Targeted/ Location Sensitive Marketing
- Smart Grid Control
- Geo Fencing
Summary

- WSO2 CEP
- WSO2 BAM
- **WSO2 ML**
CONNECTED WORLD
TRACK
26 MARCH
SALA SALVADOR

10:50-11:40
PATTERN DRIVEN ARCHITECTURE

11:55-12:45
SECURING THE INSECURE

14:15-15:05
CREATING AN API CENTRIC ENTERPRISE

15:35-16:25
NEXT-GEN APPS WITH IOT AND CLOUD

16:40-17:30
PANEL: BUILDING TOMORROW’S ENTERPRISE: REPORTS FORM THE GROUND WARS
YOU ARE INVITED TO THE WSO2 CONNECTED WORLD PUB

26 MARCH
7:30 P.M. ONWARDS
Obrigado ....!

Score your enterprise goals!
March 27-29 | Brazil

Connect : @asankama | asankaa AT wso2.com | http://asanka.abeyasinghe .org