Intel Big Data Analytics

Physics Data Analytics and Data Reduction with Apache Spark

CERN openlab Open Day

Evangelos Motesnitsalis

21/09/2017

Contents

1. Intel Big Data Analytics
2. First Use Case – Physics Analysis
3. Motivation and Vision
4. Collaboration Members
5. Progress
    - Read Files in ROOT Format directly from Spark
    - Access files stored in EOS directly from Spark/Hadoop
6. Demo
7. Future Steps

Intel Big Data Analytics

The project aims at helping and optimizing the Analytics Solutions of CERN in the areas of:

- Data Integration
- Data Ingestion and Transformation,
- Performance, Scalability, and Benchmarking
- Resource Management
- Data Visualization
- Hardware Utilization

First Use Case – Physics Analysis

What is the Goal?

"to perform Physics Analysis and Data Reduction over data acquired from the CMS Experiment with Big Data Technologies"

Motivation and Vision

Why Data Analytics and Reduction with Spark?

- Investigate new ways to analyse physics data and improve resource utilization and time-to-physics

- Adopt new technologies widely used in the industry
  - Open the HEP field to a larger community of data scientists
  - Improve chance of researchers on the job market outside academia

Collaboration Members

Who is participating?

- CERN IT Department (openlab and IT-DB)
- The CMS Experiment
- Intel

Progress

What has been done to date?

Two Main Challenges have been solved:

1. Read files in ROOT Format using Spark

2. Access files stored in EOS directly from Hadoop/Spark

We now have fully functioning Analysis & Reduction examples tested over CMS Open Data (1 TB)

1. Read Files in ROOT Format directly from Spark

DIANA-HEP developed “spark-root”

A library based on root4j

Spark can read ROOT TTrees and infer their schema

Root files are imported to Spark Dataframes/Datasets/RDDs

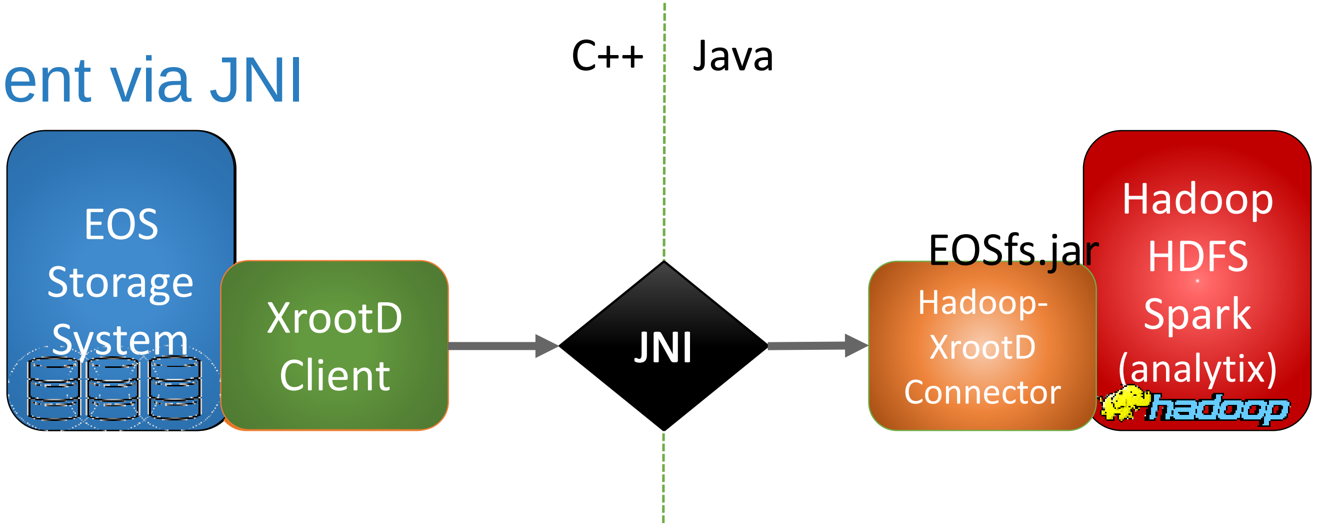
2. Access files stored in EOS directly from Spark/Hadoop

We developed “Hadoop-XrootD Connector”

A library that connects to the XrootD client via JNI

It reads files on EOS directly

(no need for import/export)



Currently working with 30% the speed of direct HDFS access for a READ operation

Lightning Demo

```
scala> val empty = Bin(40, 0, 100, {x: Float => x})
empty: org.dianahep.histogrammar.Binning[Float,org.dianahep.histogrammar.Counting,org.dianahep.histogrammar.Counting,org.dianahep.histogrammar.Counting,org.dianahep.histogrammar.Counting] = <Binning num=40 low=0.0 high=100.0 values=Count underflow=Count overflow=Count nanflow=Count>

scala> val histo = muons.as[Seq[Float]].flatMap({case x => x}).rdd.aggregate(empty)(new Increment, new Combine)
recoMuons_muons__RECO_
[Lorg.apache.spark.sql.sources.Filter;@798f8f25
17/09/20 15:06:45 WARN ClosureCleaner: Expected a closure; got org.dianahep.histogrammar.Increment
17/09/20 15:06:45 WARN ClosureCleaner: Expected a closure; got org.dianahep.histogrammar.Combine
histo: org.dianahep.histogrammar.Binning[Float,org.dianahep.histogrammar.Counting,org.dianahep.histogrammar.Counting,org.dianahep.histogrammar.Counting,org.dianahep.histogrammar.Counting] = <Binning num=40 low=0.0 high=100.0 values=Count underflow=Count overflow=Count nanflow=Count>

scala> histo.println
                      0                                                       8869.30
                      +-----------------------------------------------------------+
underflow        0    |                                                           |
[  0   ,  2.5 )  8063 |*******************************************************    |
[  2.5 ,  5   )  5173 |**********************************                         |
[  5   ,  7.5 )  1629 |***********                                                |
[  7.5 ,  10  )  379  |***                                                        |
[  10  ,  12.5)  130  |*                                                          |
[  12.5,  15  )  60   |                                                           |
[  15  ,  17.5)  26   |                                                           |
[  17.5,  20  )  19   |                                                           |
[  20  ,  22.5)  8    |                                                           |
[  22.5,  25  )  5    |                                                           |
[  25  ,  27.5)  4    |                                                           |
[  27.5,  30  )  7    |                                                           |
[  30  ,  32.5)  2    |                                                           |
[  32.5,  35  )  3    |                                                           |
[  35  ,  37.5)  2    |                                                           |
[  37.5,  40  )  1    |                                                           |
[  40  ,  42.5)  1    |                                                           |
[  42.5,  45  )  3    |                                                           |
[  45  ,  47.5)  1    |                                                           |
[  47.5,  50  )  0    |                                                           |
[  50  ,  52.5)  0    |                                                           |
[  52.5,  55  )  1    |                                                           |
[  55  ,  57.5)  2    |                                                           |
[  57.5,  60  )  0    |                                                           |
[  60  ,  62.5)  0    |                                                           |
[  62.5,  65  )  0    |                                                           |
[  65  ,  67.5)  0    |                                                           |
[  67.5,  70  )  0    |                                                           |
[  70  ,  72.5)  1    |                                                           |
[  72.5,  75  )  0    |                                                           |
[  75  ,  77.5)  0    |                                                           |
[  77.5,  80  )  0    |                                                           |
[  80  ,  82.5)  0    |                                                           |
[  82.5,  85  )  0    |                                                           |
[  85  ,  87.5)  0    |                                                           |
[  87.5,  90  )  1    |                                                           |
[  90  ,  92.5)  1    |                                                           |
[  92.5,  95  )  0    |                                                           |
[  95  ,  97.5)  0    |                                                           |
[  97.5,  100 )  0    |                                                           |
overflow         7    |                                                           |
nanflow          0    |                                                           |
                      +-----------------------------------------------------------+

scala>
```

```
.pt_")
inder.class]
root")}
)-E41F1318174C.r
setting 'spark.d
.tyBlock_: int .
struct<hash_:st
```

Future steps

Hadoop-XrootD Connector:

- Add GRID Authentication alongside with Kerberos
- Introduce optimizations and tuning to decrease the performance gap between EOS and HDFS

spark-root:

- Implement the ability to write to ROOT files
- Add IO statistics support

Investigate scaling behavior for larger input (goal is 1 PB)

Leverage Intel® CoFluent™ Technology for Big Data simulation based cluster level optimization

Investigate the possibility to use Spark over Openstack

Work with Intel and the Beams Department for their analytics use cases on Accelerator Controls

Questions?

emotes@cern.ch