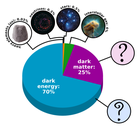
Big Data helps particle physicists to concentrate on science

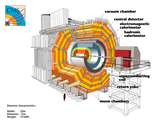
Matteo Cremonesi(**), Oliver Gutsche(**), Bo Jayatilaka (**), Jim Kowalkowski(**), Cristina Mantilla(**), Jim Pivarski(*), Saba Sehrish(**), Alexey Svyatkovskiy(*)
(*) Princeton University (**) Fermilab

Abstract

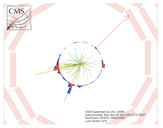
In this poster, we evaluate Apache Spark for High Energy Physics (HEP) analyses using an example from the CMS experiment at the Large Hadron Collider (LHC) in Geneva, Switzerland. HEP deals with the understanding of fundamental particles and the interactions between them and is a very compute- and data-intensive statistical science. Our goal is to understand how well this technology performs for HEP-like analyses. Our use case focuses on searching for new types of elementary particles explaining Dark Matter in the universe. We provide different implementations of this analysis workflow; one using Spark on the Hadoop ecosystem, and the other using Spark on high performance computing platforms. The analysis workflow uses official experiment data formats as input and produces publication level physics plots. We compare the performance and productivity of the current analysis with the two above-mentioned approaches and discuss their respective advantages and disadvantages.



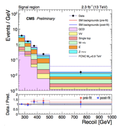
Matter_energy composition of the Universe



The CMS detector



Example of selected Collision



Publication Physics Plot

CMS Dark Matter Search

In a particle collision, Dark Matter would be produced in association with visible particles. Dark Matter particle(s) would propagate through the detector undetected while visible particles would leave signals in the CMS detector.

The signature we search for in Dark Matter production at CMS is an energy imbalance, or "missing transverse energy" associated with detectable particles.

Challenges of the analysis:

- Top quarks are relatively rare, we need to identify collisions producing top quarks with high efficiency
  - Advanced computational techniques such as artificial neural networks and boosted decision trees can greatly improve the efficiency of the top identification process
- Large backgrounds stemming from known processes will still dominate any present signal
  - Optimize the collision selection via advanced computational techniques

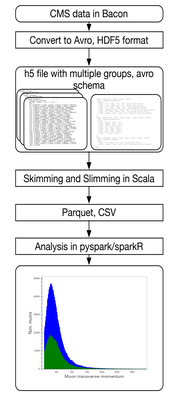
Using Spark

Spark on HPC

Apache Spark 2.0 is available on Cori and Edison at NERSC. Edison is used in the initial development and testing.

Input data: Convert n-tuples to HDF5.

- One HDF5 group per particle type (Tau, Muon, Electron)
- One HDF5 dataset per particle property in each group e.g. momentum, mass, trajectory
- Column-oriented data for faster access
- Custom HDF5 reader to read in a HDF5 group with several 1D datasets into Spark DataFrame

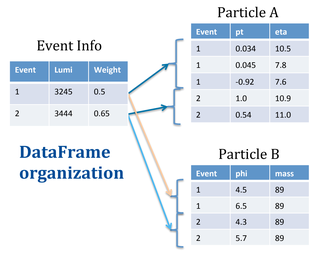


Workflow using Spark

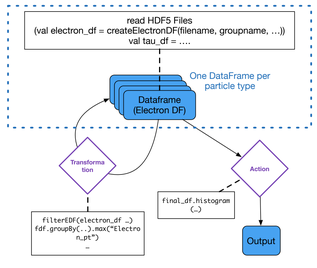
Spark operations and APIs: perform skimming and slimming on Spark DataFrames

- Use map, flatMap and filter transformations
- Use SQL queries and UDF

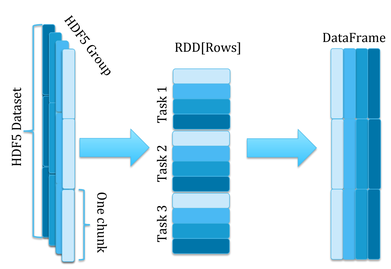
Statistical analysis and plotting: Python or R.



DataFrame organization



Flow of operations and data in Spark



Different steps of custom HDF5 reader, from HDF5 group into Spark DataFrame

Spark on Hadoop

Apache Spark on an SGI Hadoop Linux cluster consisting of 6 data nodes and 4 service nodes all with Intel Xeon CPU E5-2680 v2 @ 2.80GHz, each worker node having 256 GB.

Input data: Develop a library to convert ROOT Ttrees (the most common format in HEP) to Apache Avro row-based format readable by Spark and stored in HDFS

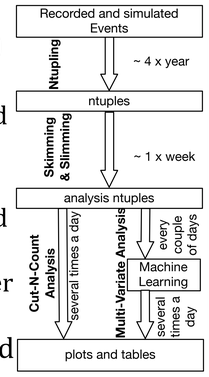
Spark operations: skimming and slimming on RDDs

- Use Spark's map, flatMap and filter transformations.
- Use Dataset and DataFrame APIs taking advantage of advanced Catalyst query optimizer and direct operations on serialized data.
- Use Apache Parquet columnar format to store intermediate results between stages of the calculation in HDFS, allowing to easily ingest data into DataFrames and Datasets

Statistical analysis and plotting: Use the histogrammar package, http://histogrammar.org

Current Approach

1. N-tupling
   - Recorded events: Analysis Object Data (AOD) event format (400kB/event).
   - 5 x 10^8 simulated events for backgrounds and signal (200 TB)
   - Processing time: 2.8 x 10^4 CPU hours Output: ROOT files in custom format (2 TB), event and physics objects information stored in vectors and float types, and fixed types of particles per event (Electron, Muon, Tau)
2. Skimming (reduce number of events) and Slimming (reduce event content)
   - Apply analysis pre-selection
   - Output: flat n-tuples with only necessary information
3. Final analysis is performed using the flat ntuples
   - Output: publication grade plots and tables

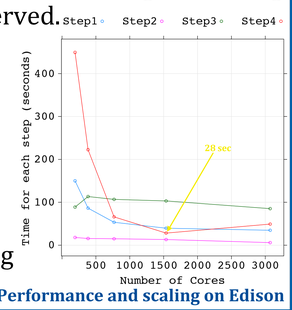


Workflow using current approach

Performance at NERSC

3072 cores, and total of 0.5 TB of data: The input data is stored in 928 HDF5 files comprised of 360 million events, which are distributed across 243 Lustre FS OSTs. The analysis operation created a histogram of transverse momentum of ~200 million electrons using 2499 Spark data partitions (tasks).

- Step 1 is the time to read HDF5 datasets into RDDs; good scaling for partitions greater than cores is observed.
- Step 2 and 3 show time to format RDDs and convert to DataFrames; a constant overhead is observed.
- Step 4 is the time to perform various cuts on the DataFrame using UDFs and operations such as aggregation, filter and joins. Good performance and scaling when partitions are greater than cores.



Performance and scaling on Edison

Conclusion

The goal of our exploratory study is to shorten *time to physics* using Spark

- Observed scalability, task distribution and data partitioning
- Availability of pySpark and SparkR high level APIs are appealing to the HEP user community
- Encoding our workflow using Scala best practices along with the optimal use of DataFrame features is challenging
- Documentation, and error reporting needs to be better

Future Direction

- Improve the HDF5 reader interface
- Optimize skimming workflow
- Scale up to greater than TB data set

Acknowledgement This study is supported by the Department of Energy (DOE) under Contract No. DE-AC02-07CH11359. The CMS collaboration is supported by DOE, NSF, and international funding agencies. Histogrammar is supported through the DIANA project by NSF.