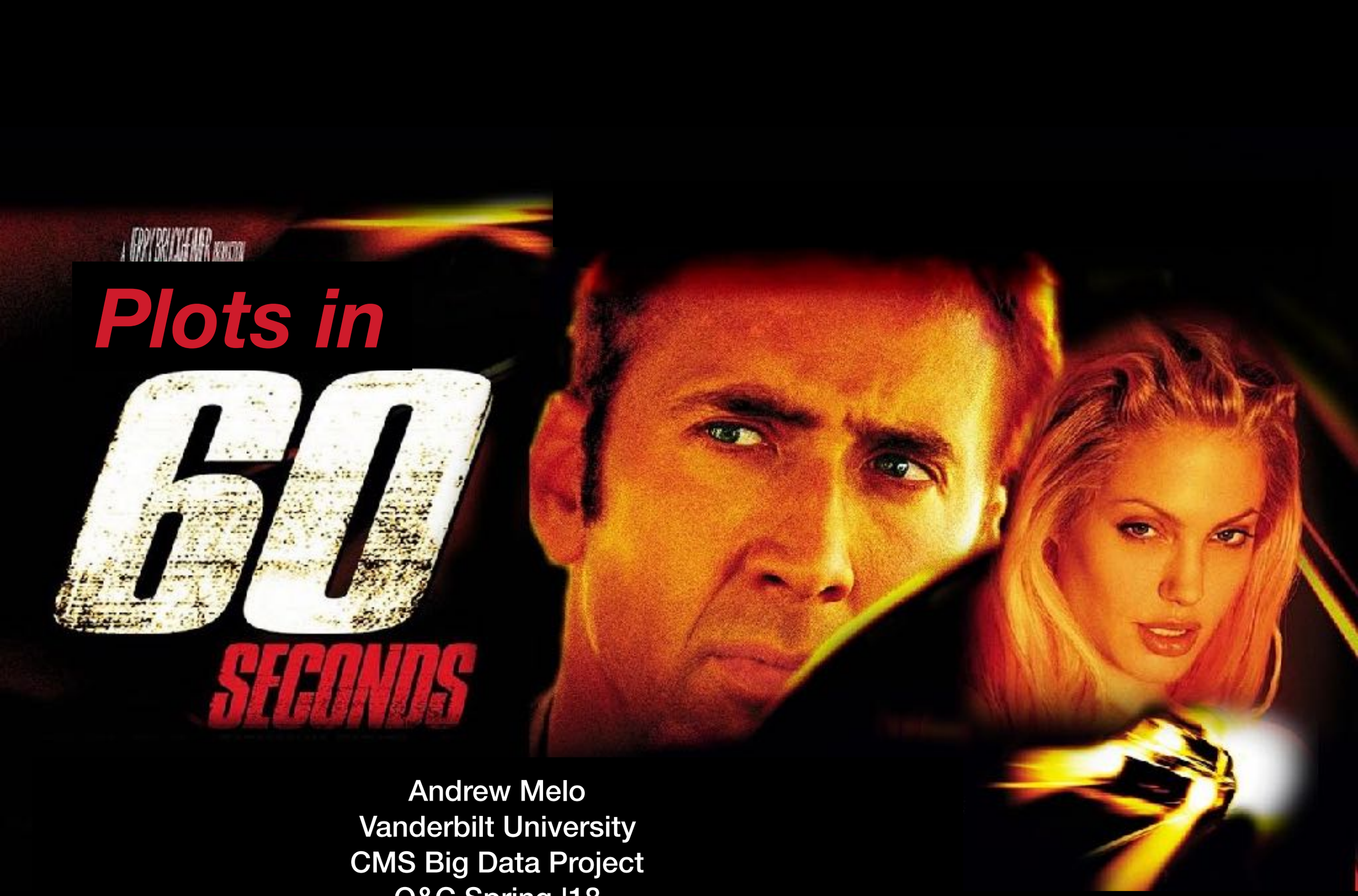
Plots in



Andrew Melo
Vanderbilt University
CMS Big Data Project
O&C Spring '18

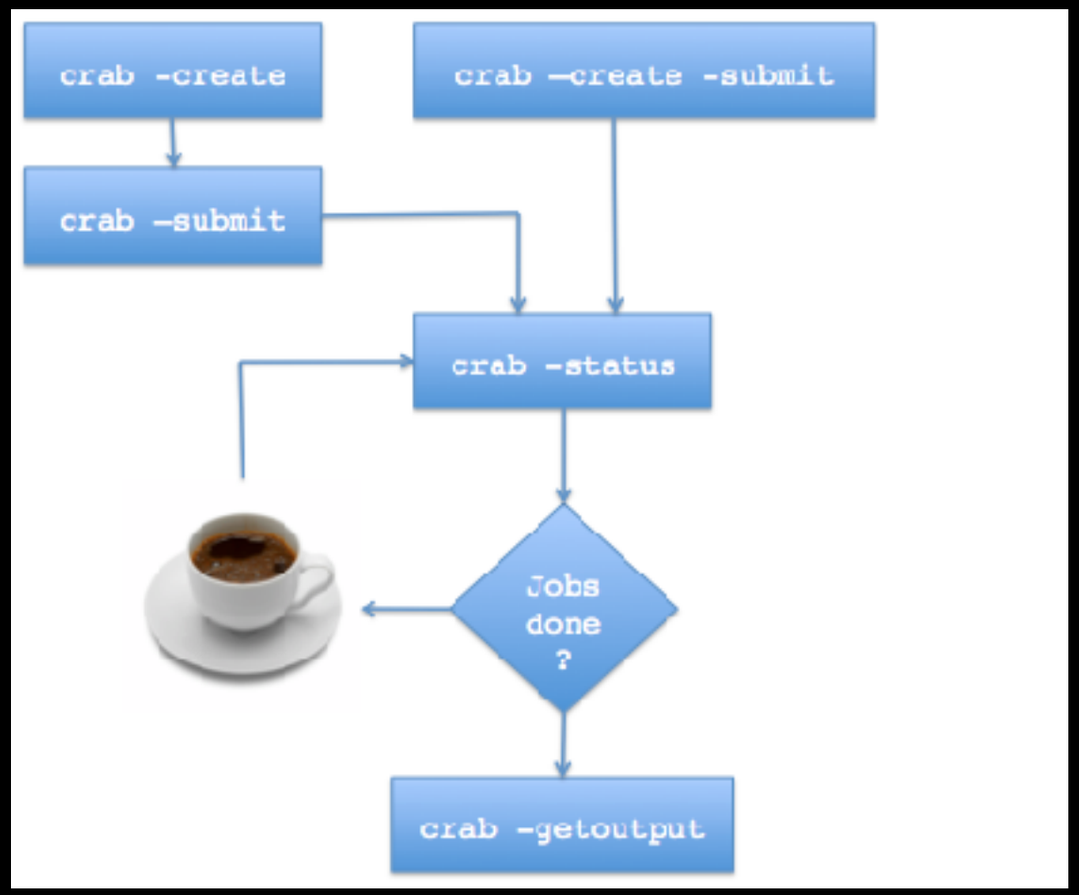
Motivation

Motivation

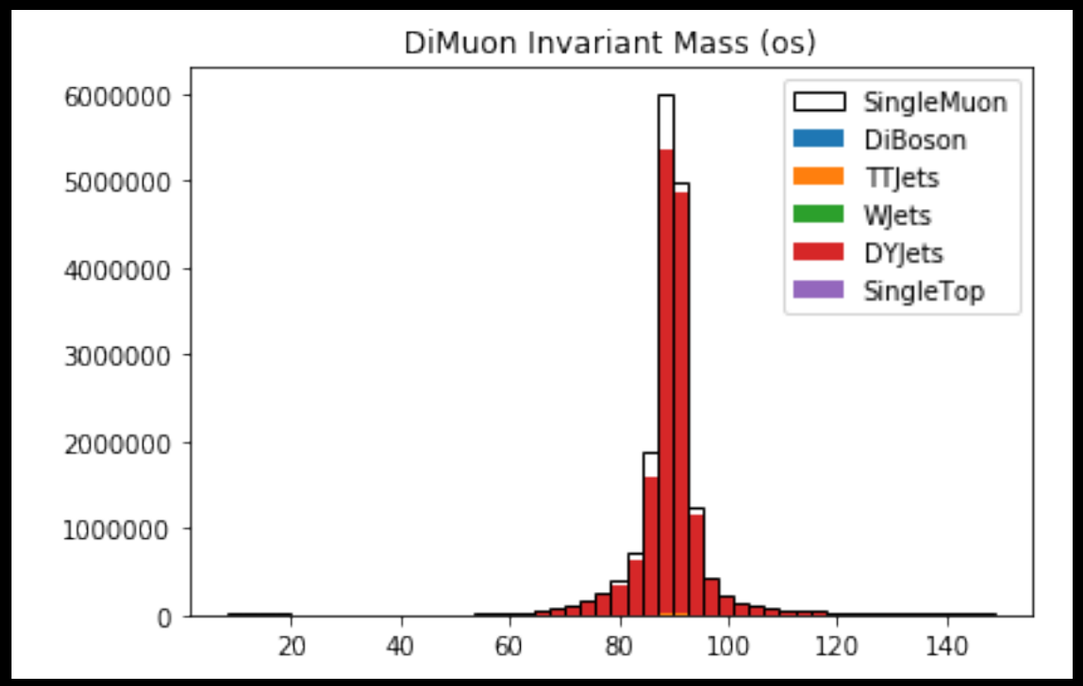
"How does that look if you increase the MET cut to 30GeV?"

Latency Hurts

- Interactive tasks are sensitive to latency.
- Analysis is often "hit enter, come back in an hour"
- Context-switching is inefficient for humans
- Can we lower the "reducible" latency?



Analysis with Spark



931M events in ~90 secs

Change a cut

New plots in ~15 secs

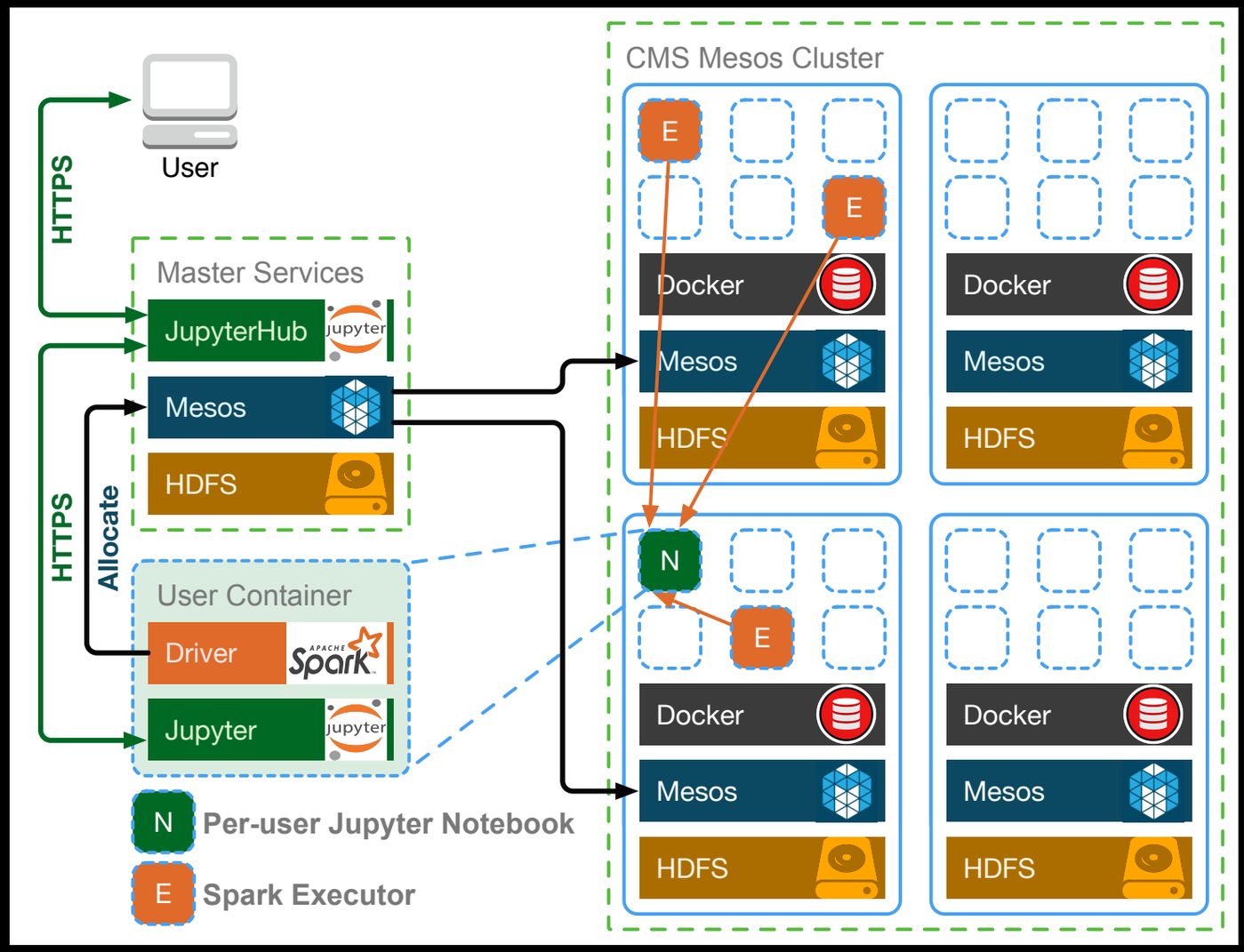
- Analysis-level non-flat ntuples
  - ~20TB total size
- Time includes the full chain
  - Core aquisition, File I/O, cuts, flattening, histogramming, plotting
- 350 cores, HDFS on spinning HDDs

Apache Spark

- "...a fast and general engine for large-scale data processing"
- Replacement of the "Hadoop" component of the Hadoop ecosystem
- DataFrames are the ~columnar data abstraction
- Operations on DataFrames produce new DataFrames
- Spark schedules an optimized DAG of operations
  - Intermediate results are elided or cached
- Code is same running locally or on a cluster
  - Spark handles splitting tasks, propagating results, retries, etc..

Spark @ Vanderbilt

- Cluster funded by internal grant
- JupyterHub frontend
- Each user gets personal container w/Jupyter Notebook
  - Frontend routes traffic
- Notebooks and Spark executors allocate from same Mesos pool

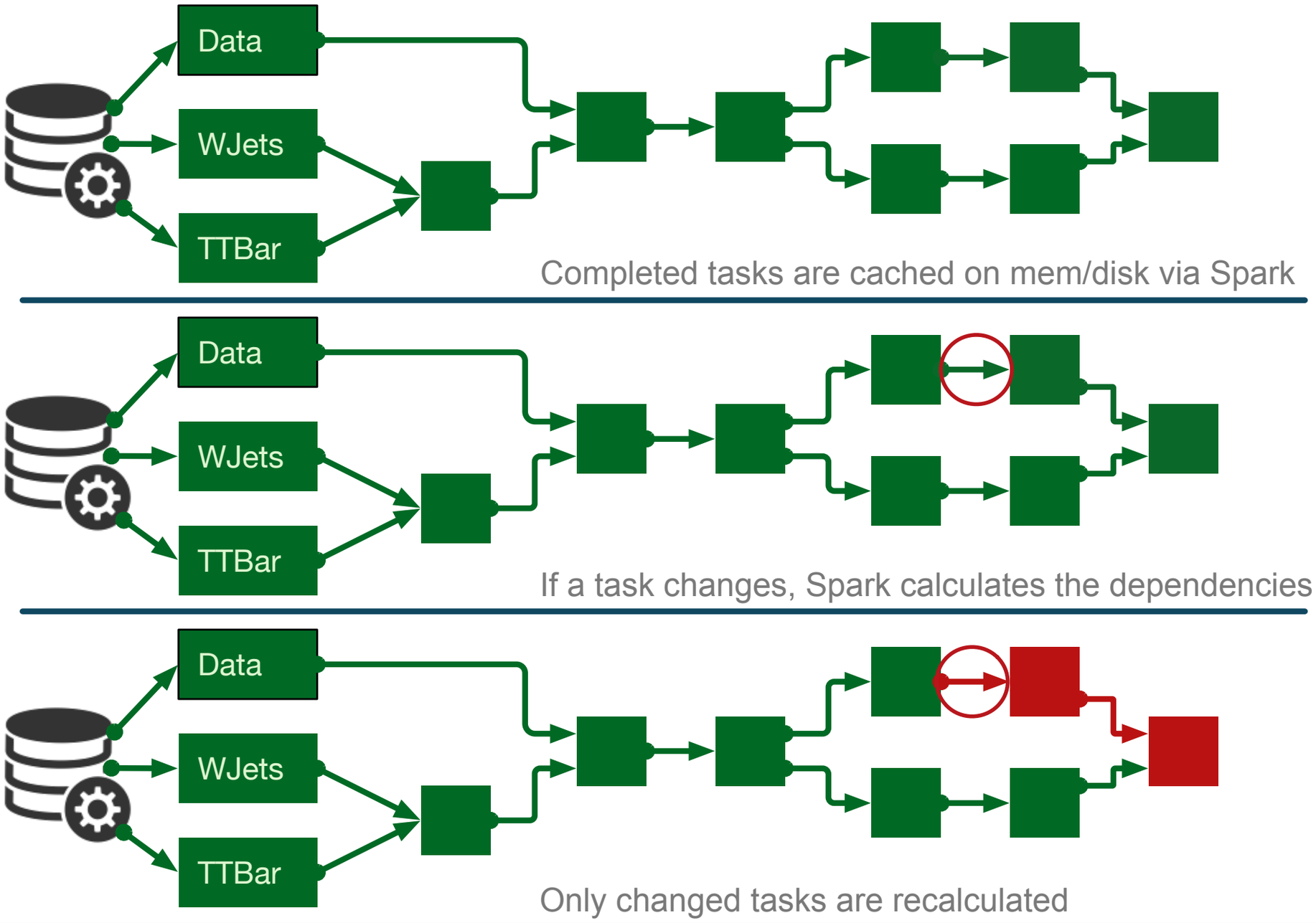


Analysis Workflow

- Load standard ROOT (edm)ntuples as DataFrames (DFs)
  - Support coming soon for XRootD access
- Use Spark to transform DFs
  - SQL-ish operations
  - Arbitrary Python/C++/Java(etc..) functions
- Aggregate DFs into histograms
- Produce plots, tables, etc.. from histograms

Tools used:

- **Spark-ROOT - ROOT in Spark**
- **Hadoop-XRootD - XRootD FS support for Hadoop**
- **Histogrammar - Data aggregation**
- **Matplotlib - Python-based plotting**



Iterative Analysis

Spark only recalculates necessary parts of the DAG.
Once the inputs are cached, changes go quickly

Short-Term Goals

1. Reproduce AN-17-142 with Spark+Jupyter
2. Develop a more comprehensive tutorial*
3. Get ≥1 "not savvy" user
    - E.G. A technically savvy user who is unfamiliar with Spark

*** There was a tutorial at the LPC in Dec: https://github.com/FNALLPC/spark-hats**

Progress

- We've duplicated (abridged) analysis results in Spark
- Still needs polish
- Lots of boilerplate
- Plots aren't 100% nice
- Slowly finding new needs and developing features

Next Steps/Questions

- Finish converting AN-17-142 2x
  - Short demonstration of one CR ✓
  - Full reproduction of plots/tables ✗
- Streamline EOSConnector deployment
- Find a ~~victim~~ user

- Write a library to replace the (substantial) boilerplate?
- How do we make notebooks interoperable between sites?
  - Storage locations, etc
- How should we best advertise these resources to users?
- How can we publish these results?
  - Uses CMS data

Interested?

- Bi-weekly meeting Weds @ 1600h CERN
- Slack @ cmsbigdataproject.slack.com
- Contact Andrew Melo for access to Vanderbilt's Cluster
- Looking for Python/Java/Scala developers
    - Plenty of projects for all skill levels