# Using a High-performance, Column-store Database for Particle Physics Analysis

Matthew Link, Kevin Lannon, Jason Slaunwhite
Department of Physics, University of Notre Dame

## Introduction

ROOT, a data analysis framework, is the particle physicist's tool for analyzing data. It allows researchers to make progressively more complex selections of the data. At each stage, a criteria is given to select interesting subsets of the data while the rest is discarded. The early criteria is simple and known to work. It does not have to be repeated. The late criteria is where the physics research happens. It is where researchers apply their knowledge and creativity to push the bounds of physics. However, no one knows if the late criteria will work. It has to be repeatedly tested and revised. Each test takes significant time as ROOT spins through the data. The researchers wait. Time spent spinning through data is time not spent thinking about physics. For physics researchers, this is not ideal. The private sector has faced similar data problems and has developed tools to handle them. Our research focused on employing a commercially available, high-performance column-store database to reproduce particle physics analysis done in ROOT. We reproduced large skims of the data to produce smaller data sets.

## Particle Physics

Particle physicists want to answer questions about the fundamental pieces of the universe. To do this, they collide protons together at very high speeds, breaking the them apart.

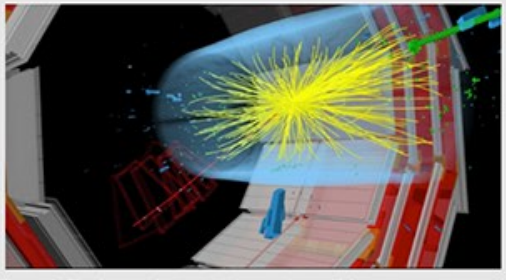

Figure 1. The image shows a proton-proton collision.
Tom McCauley/CMS

The result of each collision is a single event and each event has many reconstructed objects. Using those objects, physicists can construct a story about what happened in the event. Those objects all have variables associated with them that tell how interesting the collision was. Physicists create many collisions because interesting ones are rare. As a result, a massive amount of data is generated.

## Using ROOT and the database

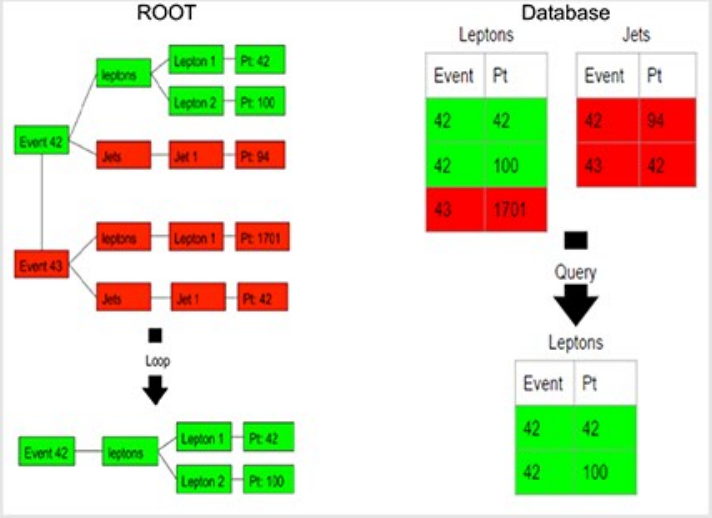


Figure 2. The figure on the left shows how data analysis is done in ROOT. The data is stored in a tree which contains all of the events and their objects. To identify interesting events, a loop is constructed in C++ to run over the events and objects. Leptons and jets are examples of the objects. The figure on the right shows how data analysis is done in the database. All of the objects are stored in tables and SQL queries and run against the tables to find interesting events.

## Comparing ROOT and the database

| | ROOT | database |
|---|---|---|
| Who wrote the program? | Moderately experienced graduate student | Novice Vertica and SQL undergraduate |
| Effort put into optimization? | None. Both have the potential to be tuned to run faster but we aimed to measure how a standard, non-expert user would use them. | |
| What was measured? | Total runtime of the program for varying data sizes | |

Figure 3. This table shows how we compared the performance of ROOT and the database.

## Results

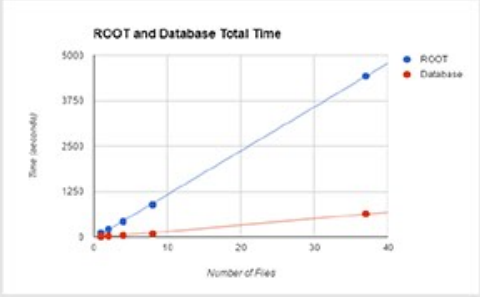


Figure 4. The graph shows the total runtime for ROOT and the database. Each data point is an average of multiple runs. The number of files containing data was increased to test how the programs scaled with more data. The times for ROOT fall right on a line which suggests that it scales linearly and that the times are valid.

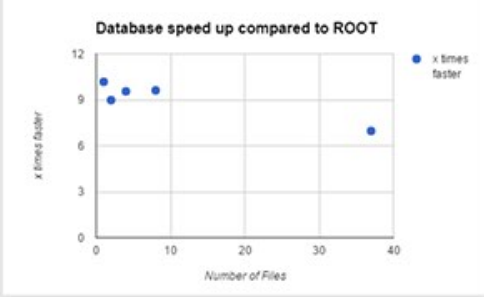


Figure 5. The graph shows the speed up found by using the database for each number of files tested. The database was about 10x faster for a low number of files. At 37 files, it was about 7x faster. Under the hood, the database attempts to make a good plan for accessing the data. The large increase in data may have been more than it could fit into memory so it chose a different plan that was not as fast.

## Acknowledgements

This research was funded by NSF Award Number: 1560363