

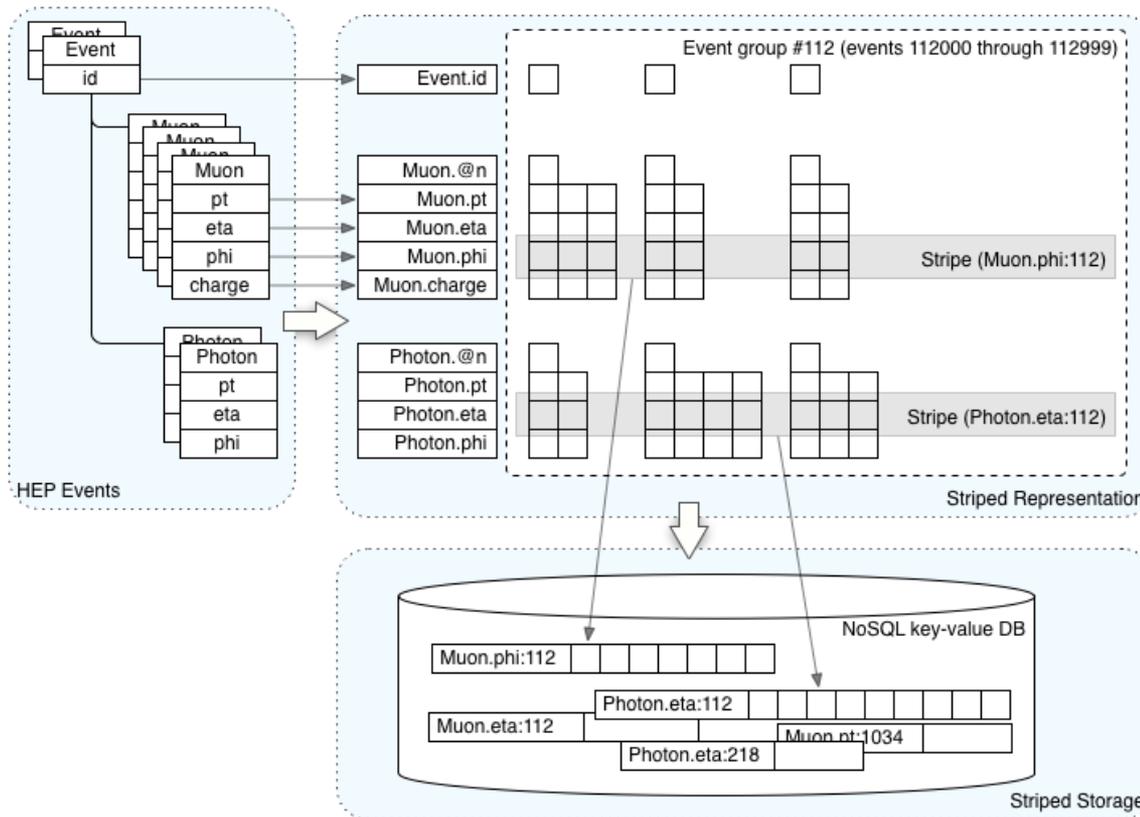
STRIPED DATA STORAGE AND ANALYSIS PLATFORM

Igor Mandrichenko

EAG meeting

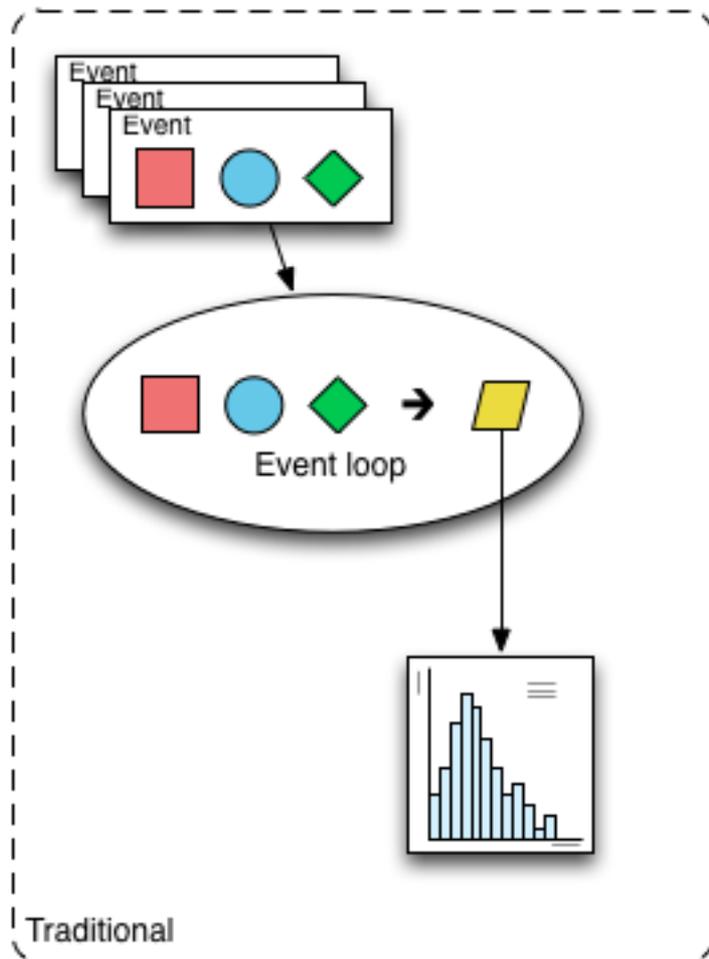
9/20/2017

Striped Data Representation



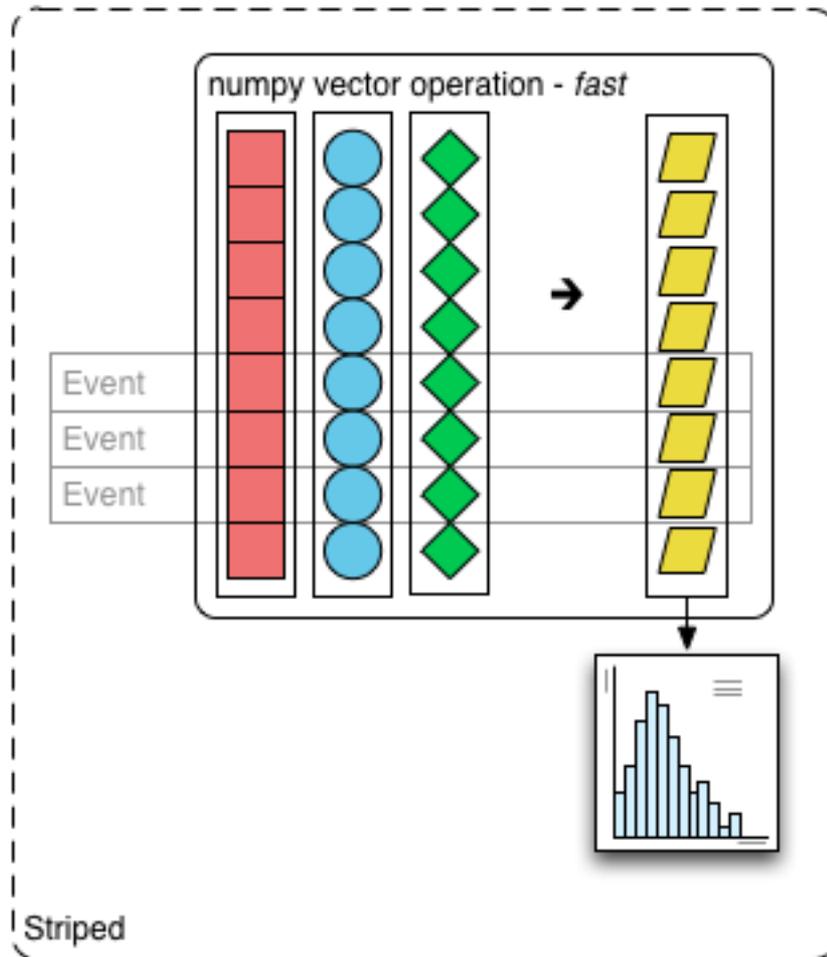
- Columnar representation
- Scales easily to 1000 columns and beyond
- *Stripe* is the unit of representation one column for 1-10K rows
- Internal format is numpy array
- Any key-value storage can be used as the backend
- Distributed NoSQL backend for scalability

Traditional, event loop analysis



- Load next event (object) as a tuple
- Calculate missing quantities
- Add them to histogram
- Repeat

Striped analysis

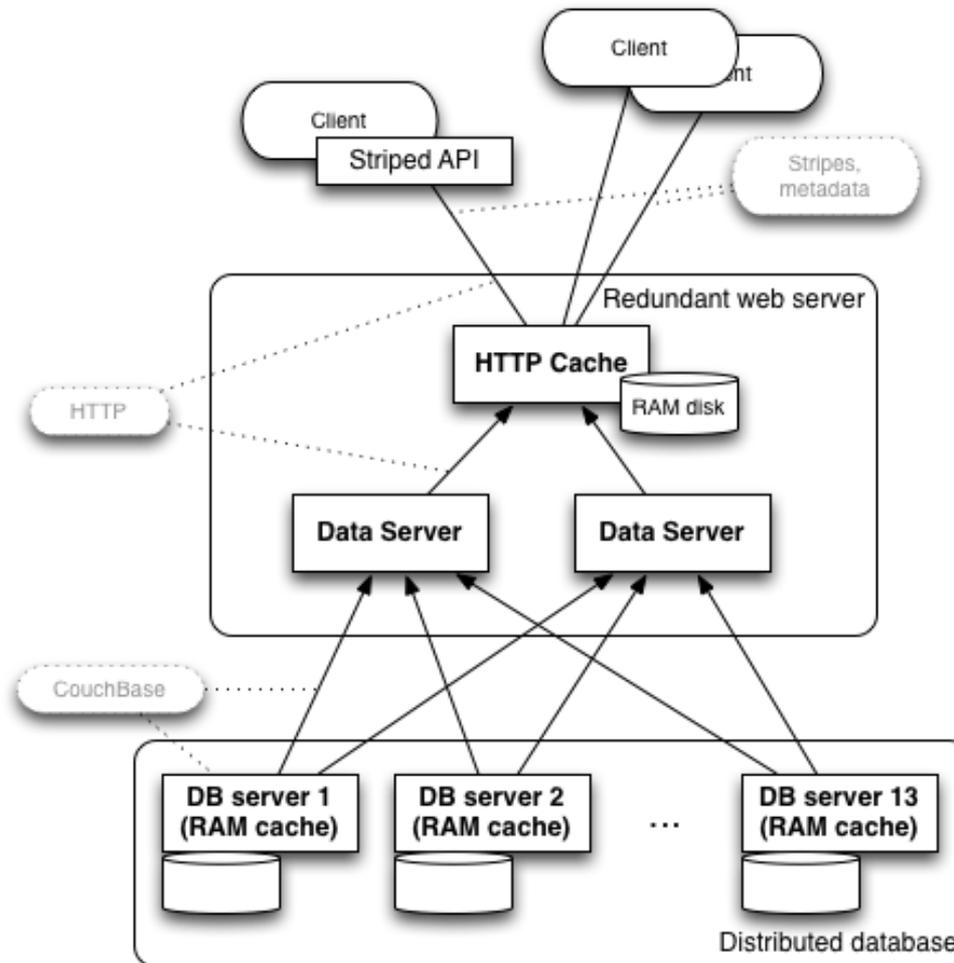


- Load next event (object) group
1-10K events (objects)
- Calculate missing quantities
as vectors, using numpy
- Add them to histogram
- Repeat

Advantages of striped representation

- Only needed columns have to be downloaded
- Downloaded stripe is immediately useful via numpy without additional decoding
- Stripes are cacheable
 - Same repeating analysis processes
 - Different analysis processes using same pieces of data
- Some calculations can be performed using numpy on stripes
 - E.g.: $\text{Muon.p} = \text{Muon.pt} * \cosh(\text{Muon.eta})$

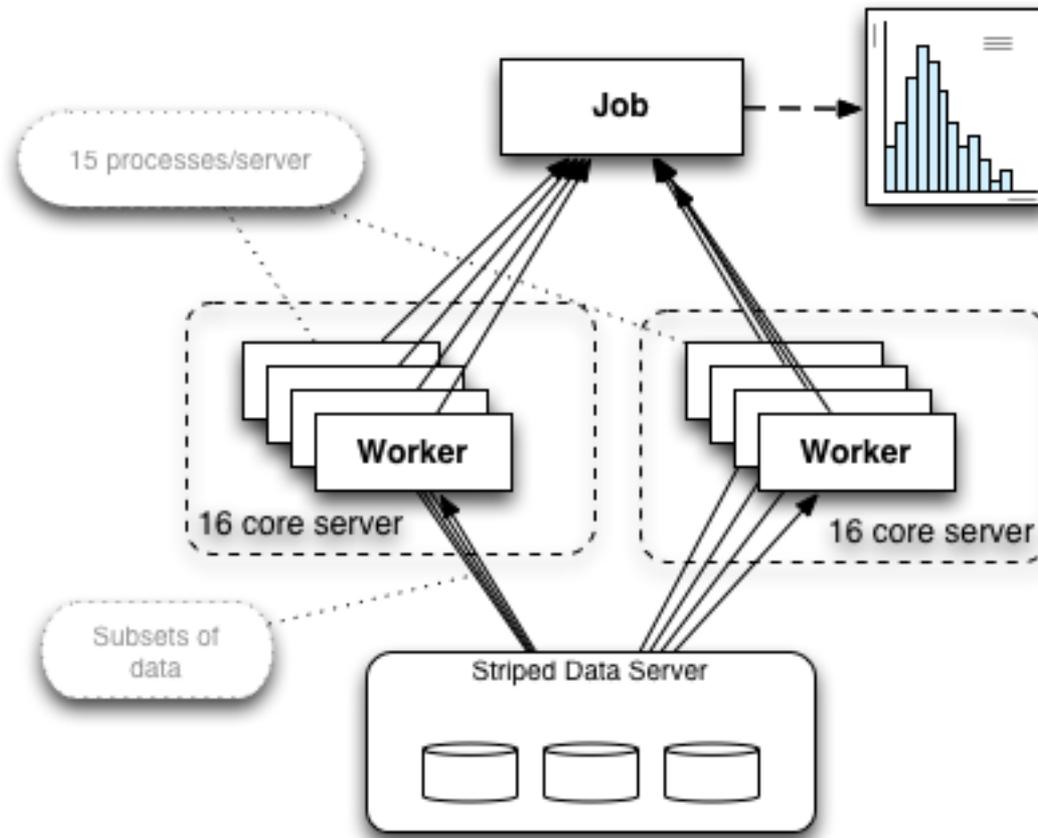
Striped Data Server Architecture



CMS Dark Matter Search Dataset

- Built Striped Storage, 13 old farm nodes + 2 production web server computers with SSD cache
- 35 datasets, most MC, some read CMS data, 1-82M events/data set
- 1.95 TB on disk, ~163M stripes
- Most datasets stored with 1000 events/group (stripe)
 - Not most efficient
 - The most efficient seems to be 10,000 events/group
- Some are stored at 20,000 events/group
- <http://dbdata0vm:9091/striped/app/datasets> (takes time to load!)

Demo Structure



User Code Structure

- Worker module
 - Define list of columns to be used
 - Typical analysis uses fraction of available columns
 - Define additional columns to be calculated
 - Using vector operations
 - Loop through events and emit compiled data
- Job module
 - Collect data from workers
 - Compile into histogram

DEMO

Conclusion

- Striped data representation developed
 - Fast, scalable, efficient direct access to large amounts of data
 - Capable of representing data structures
 - Complex - HEP event
 - Simple - relational
- Data server developed and deployed
 - CMS dark matter search dataset converted and uploaded
- Analysis platform developed
 - Vector calculations over large (1K-10K) event (object) groups
 - Python based
 - Interactive IPython, Jupyter notebook interface available
- With 30 cores, capable of processing
 - 1M events/second