CMS Analysis Workflow

Sudhir Malik

Fermilab/University of Nebraska-Lincoln, U.S.A.

CMS Software

- CMS software (**CMSSW**) based on Event Data Model (**EDM**) - as event data is processed, products stored in the event as reconstructed (RECO) data objects (each TTrees in ROOT represents an object, C++ container)
- Structure flexible for reconstruction, not usability, information related to a physics object (e.g. tracks), stored in a different location (e.g. track isolation)
- Data processing steered via Python job configurations

- CMS also provides a lighter version of full CMSSW called **FWLite (FrameWorkLite)**
- **FWLite** is plain ROOT + data format libraries and dictionaries capable to read CMS data + some basic helper classes

Guidelines and Boundary conditions

CMS has set some boundary conditions in the analysis structure, given the geographical spread of the collaboration, high cost of the program and the potential of major discoveries and breakthroughs.

- Physics results are of a very high quality in nature
- Understood and largely reproducible by other collaboration members
- Reproducible long after the result is published
- Datasets and skims selected on the basis of persistent information and the retention of provenance information
- While most of the tools are officially blessed, the analysis model has enough flexibility to support those analyses that require special datasets and tools
- Software platform is Scientific Linux on which CMS develops and runs validation code
- Physics objects and algorithms approved by the corresponding Physics Object Groups (POGs, experts in physics object identification algorithms in CMS)
- The origin of the samples should be fully traceable by the provenance information
- Analysis code on the user sample fully reviewable by Physics Analysis Groups (PAGs, experts providing physics leadership in CMS)

Flexible and Distributed Computing

Grid to provides data access to all the collaborators

CRAB provides a user front-end to submit physics analysis jobs to all CMS datasets within the data locations driven by the distributed computing structure

Tier 0 (T0) at CERN

- Primary reconstruction of raw data
- First calibration constants
- Jet energy corrections

Tier 1 (T1) Each in 7 countries

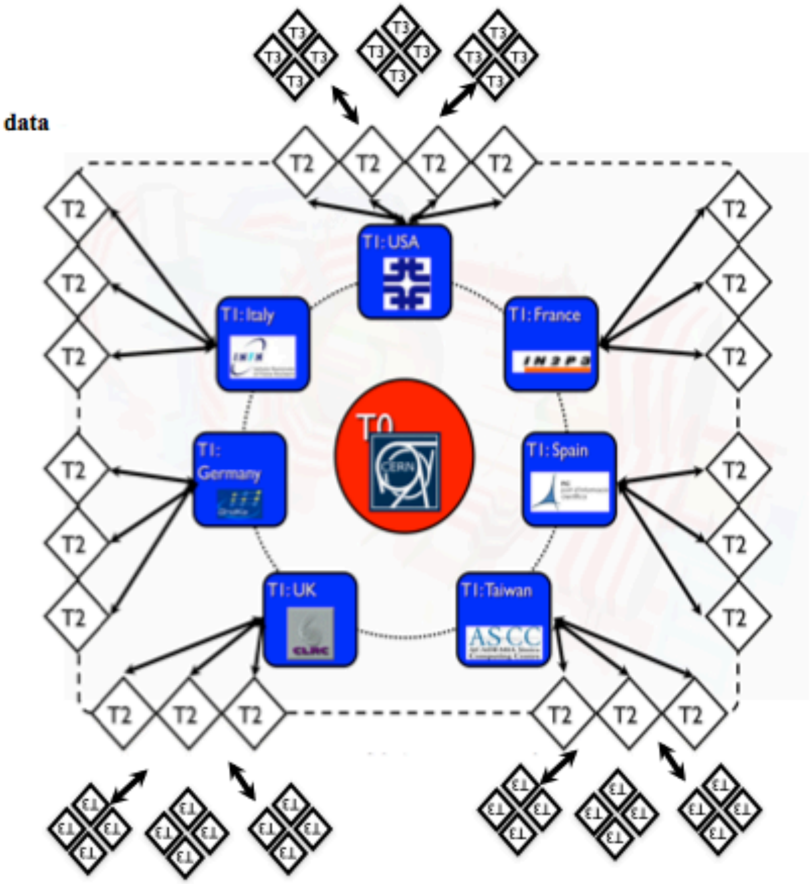
- Data reprocessing
- Data Selection
- Central Skims

Tier 2 (T2) Each in 7 countries

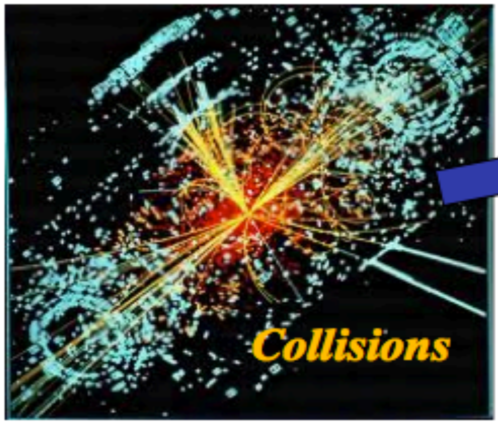
- Physics group data storage
- Produce simulated data
- Primary source user analysis
- User defined event content
- Produce plots

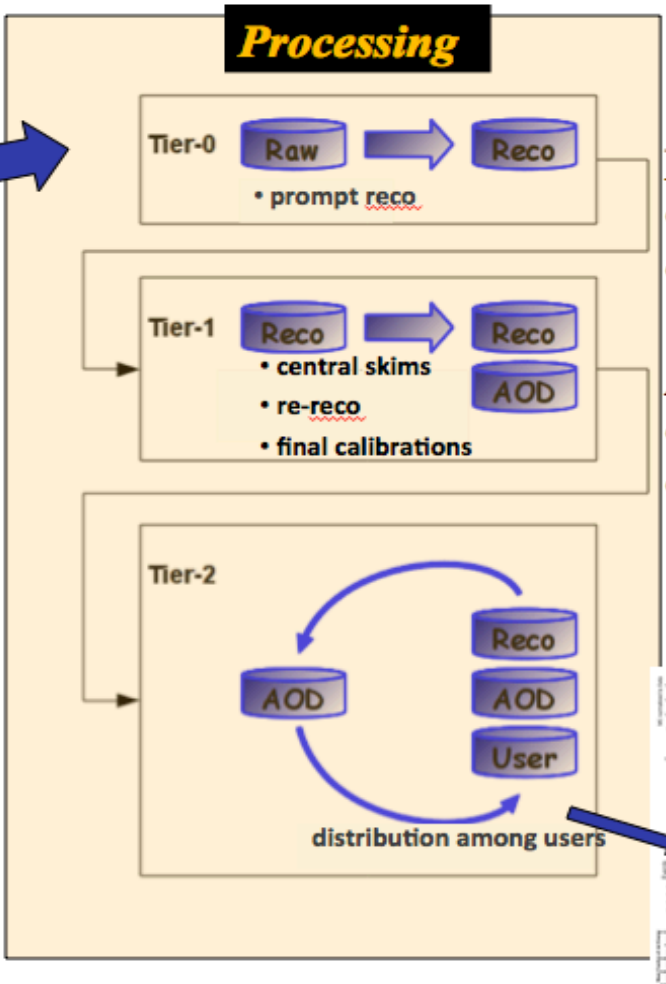
Tier 3 (T3)

- Source user analysis
- User defined event content
- Copy reduced dataset to your favorite machine



Data Tiers and Flow





Processing

Tier-0

Raw → Reco

- prompt reco

Tier-1

Reco → Reco

AOD

- central skims
- re-reco
- final calibrations

Tier-2

AOD

Reco

AOD

User

distribution among users

Data Tiers for Analysis

RECO (RECOnstructed)

- reconstructed objects and hits
- typical size of 500 kB/event

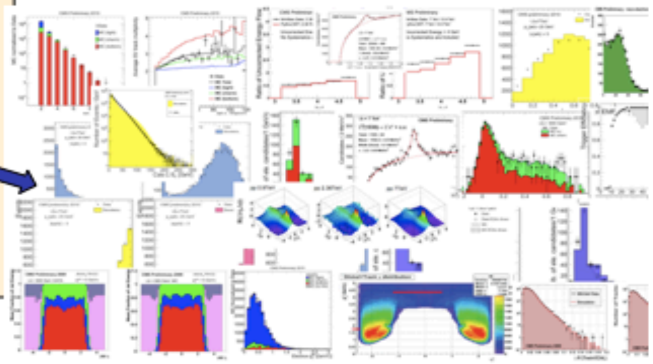
AOD (Analysis Object Data)

- smaller subset of RECO
- typical size of 140 kB/event

Data stored as ROOT files

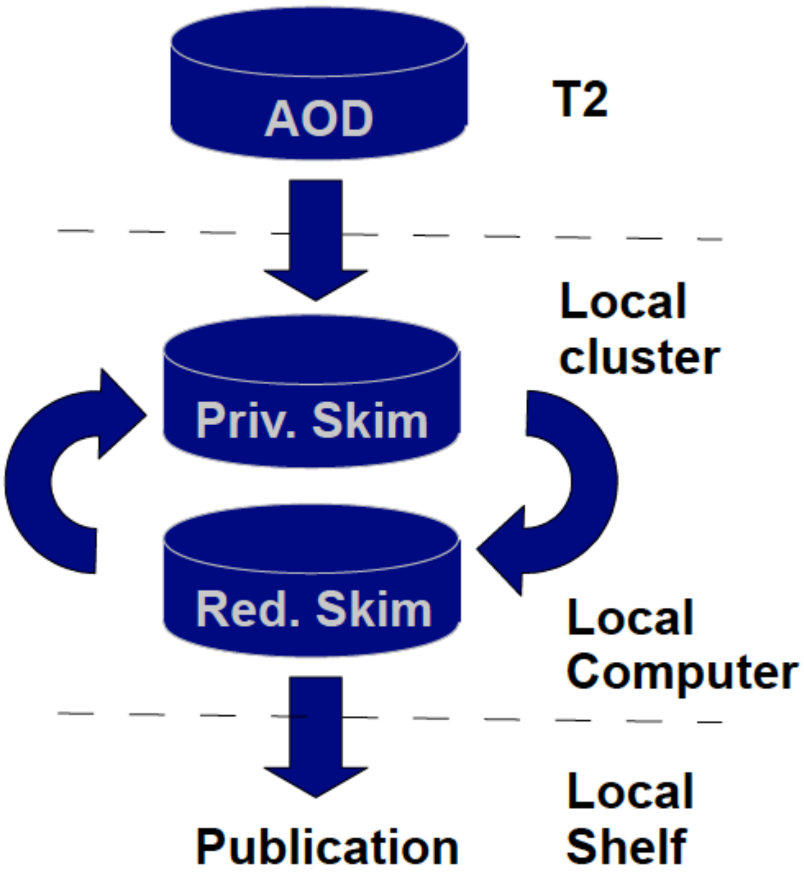
- Event info stored as TTrees
  - Standard in HEP
  - Inspected TBroswer
  - connected by smart pointers
  - Strong heirarchy
  - Minimal space overhead

Plots and results



Data Analysis Workflow

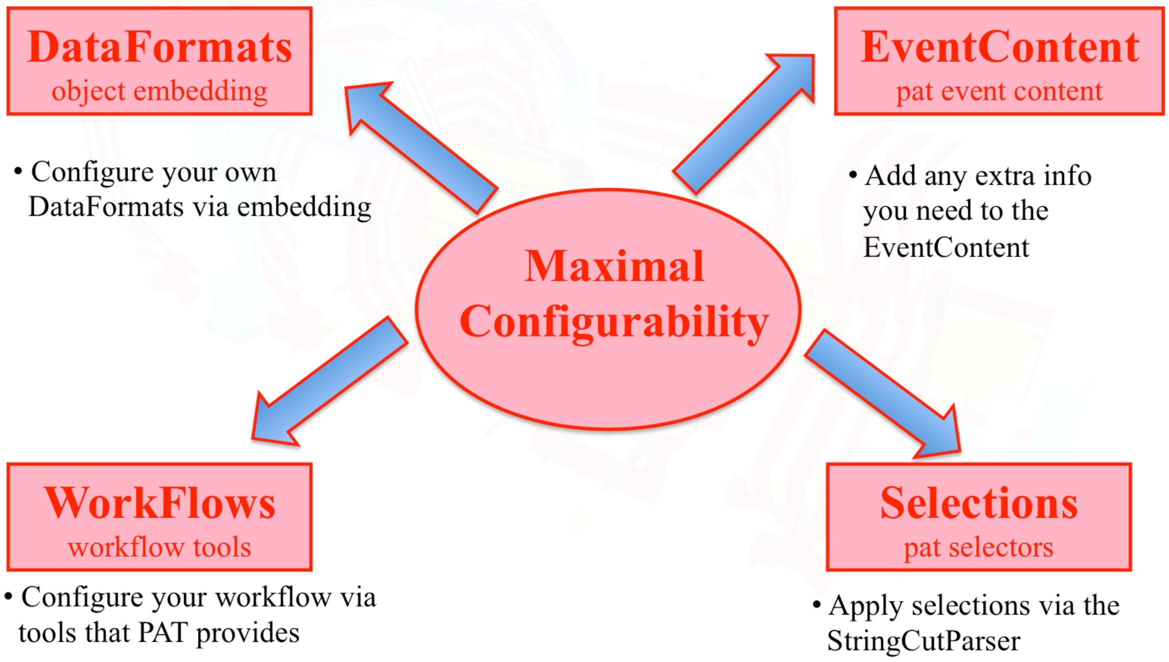
- Start from RECO/AOD on T2.
- Create private skims with less and less events in one or more steps.
- Fill histograms, ntuples, perform complex fits, calculate limits, toss toys, ...
- Document what you have done and PUBLISH



Physics Analysis Toolkit

- To ease this situation and keeping the goal of physics analysis, a special software layer called **PAT (Physics Analysis Toolkit)** was developed

- **facilitates access to event information**
- **Combines -> flexibility + user friendliness + maximum configurability**
- **provenance, uses official tools/code, one can slim/trim/drop event info**



DataFormats
object embedding

- Configure your own DataFormats via embedding

EventContent
pat event content

- Add any extra info you need to the EventContent

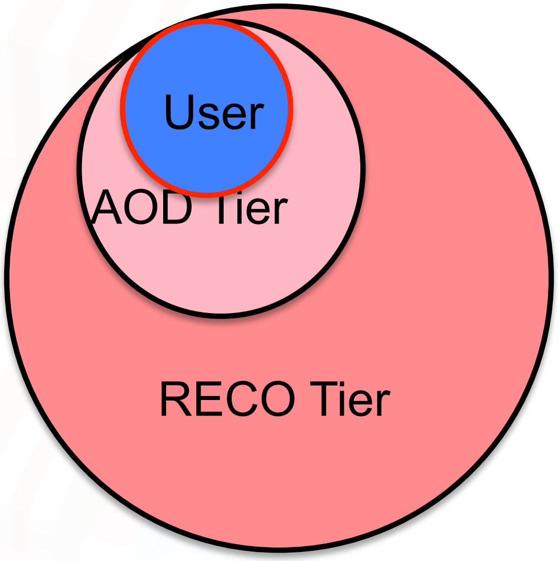
Maximal Configurability

WorkFlows
workflow tools

- Configure your workflow via tools that PAT provides

Selections
pat selectors

- Apply selections via the StringCutParser



Ways to analyze data

- Directly using RECO/AOD objects
  - Need expert knowledge to
    - know all available features
    - keep up-to-date with all POGs

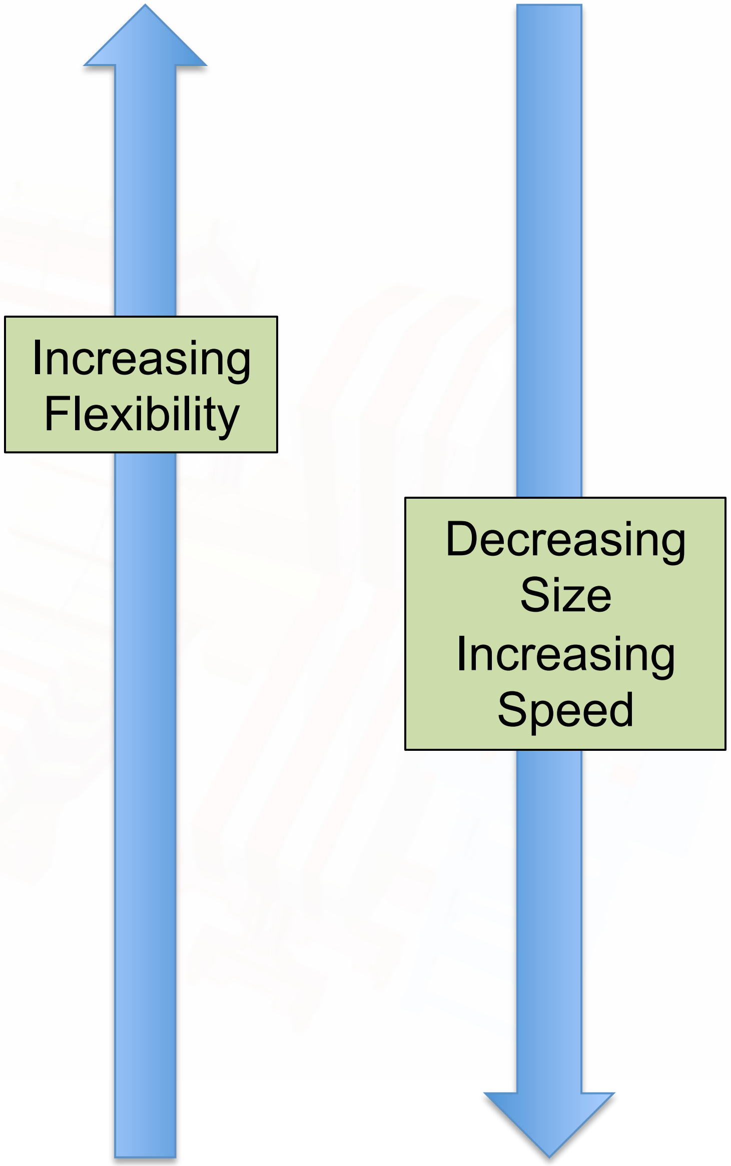
For Experts

- Producing and using PAT objects
  - All up-to-date features collected from all POGs
  - Get latest algorithms always from the same interface: the PAT objects

For Users

Ways to store data

- RECO/AOD skim
  - Full flexibility
  - Need a lot of space
- PATtuple
  - Full flexibility
  - Save space by embedding
- EDMtuple
  - Not flexible, need to know exact set of objects, variables
  - Provenance information is kept
- Tntuple
  - Give up provenance information
  - Not officially supported



PATtuple

- The challenge/problem:
  - Analyst wants all relevant information from RECO/AOD stored in an intuative and compressed way
  - RECO/AOD can only be skimmed by keeping and dropping complete branches, but not selected objects from branches

- Solution: PAT objects
  - Embedding allows to keep only relevant information
  - All relevant information from a single interface for each physics object

Data stored in form of PAT objects (C++ classes)

EDMtuple

- The challenge/problem:
  - TNtuple is the fastest way to store and plot data
  - TNtuple lacks EDM provenance information needed for reproducible analyses! e.g. for proper calculation of uminosity
  - TNtuple lacks integration with GRID software

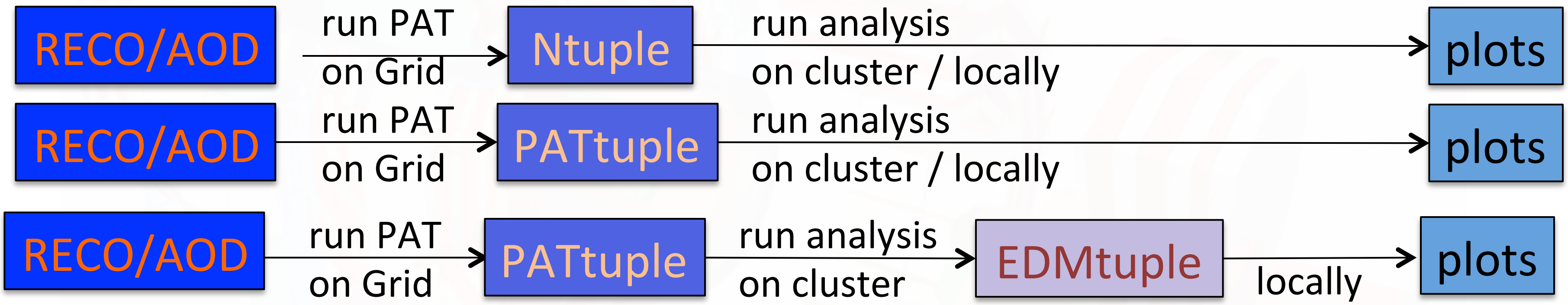
- Solution: Storage of a set of user-defined doubles in EDM Format
  - As fast as TNtuple and provenance information!

Data stored in form of a flat ntuple (vector of doubles)

Common WorkFlows

Analysis development phase:

- Keep all relevant information in a PAT tuple
- Be fully flexible for object and algorithm changes



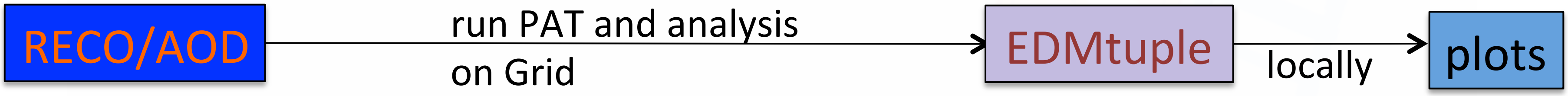
RECO/AOD → run PAT on Grid → Ntuple → run analysis on cluster / locally → plots

RECO/AOD → run PAT on Grid → PATtuple → run analysis on cluster / locally → plots

RECO/AOD → run PAT on Grid → PATtuple → run analysis on cluster → EDMtuple → locally → plots

Analysis update with more data:

- Keep all necessary information in a small EDM tuple
- Use highest performance format for fast updating of the analysis



RECO/AOD → run PAT and analysis on Grid → EDMtuple → locally → plots

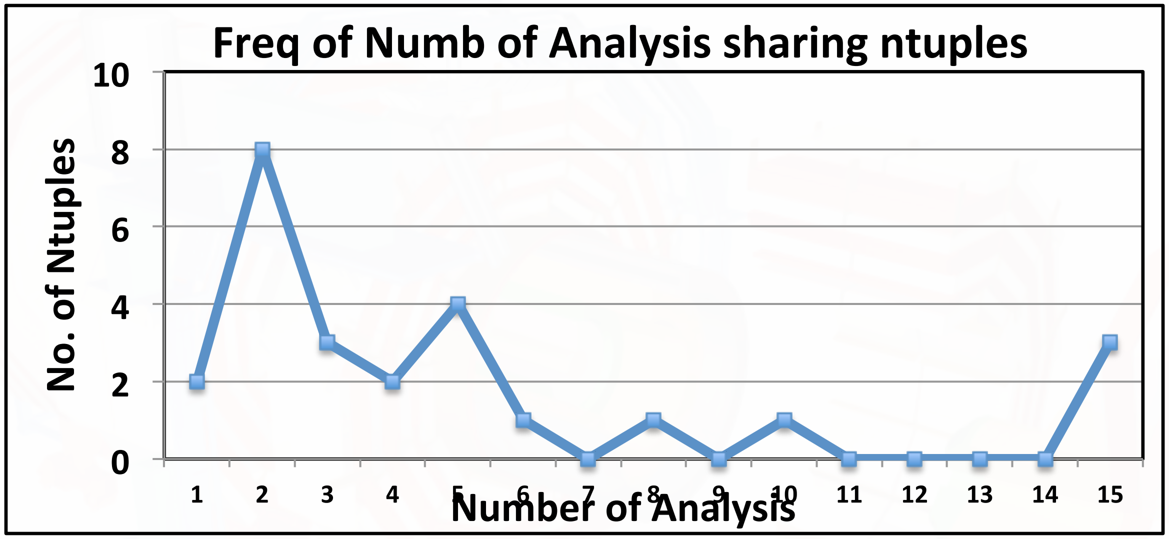
Analysis Practices

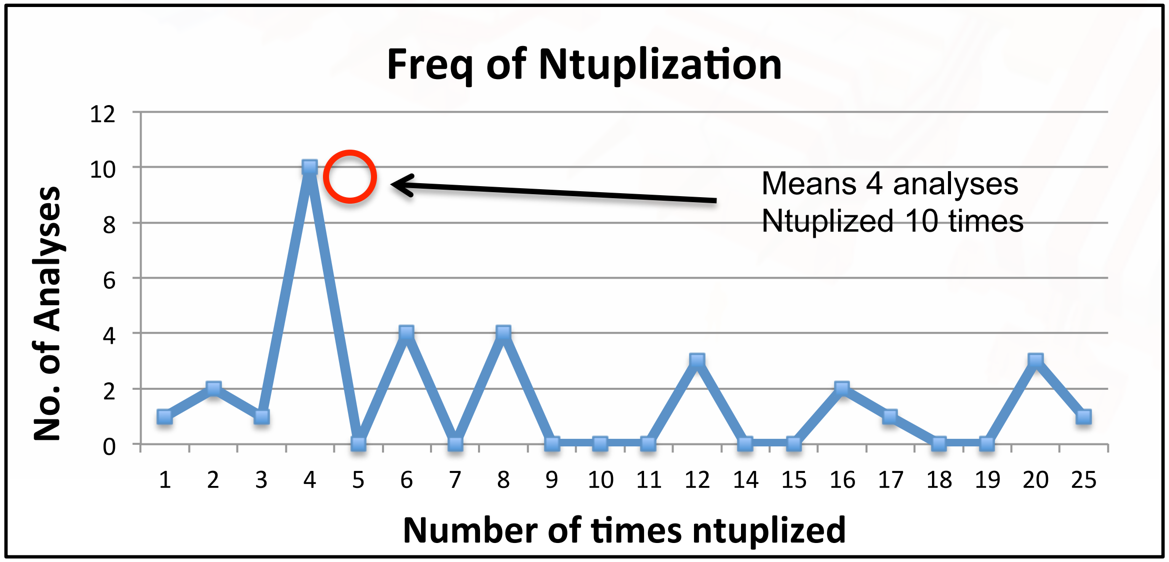
While software and physic tool determine the data preservation plan, its success depends on the analysis practices followed

- CMS is very young experiment, hence fortunate to plan data preservation
- It has a strong culture of documentation – WorkBook, SoftWareGuide
- It has strong culture of User Support
- Multiple Physics Analysis Schools per year
- All the above iron out usage of tools and software and serve as basis for long term preservation for ourselves
- In the following slides is a brief preview of these practices

n-tuple usage

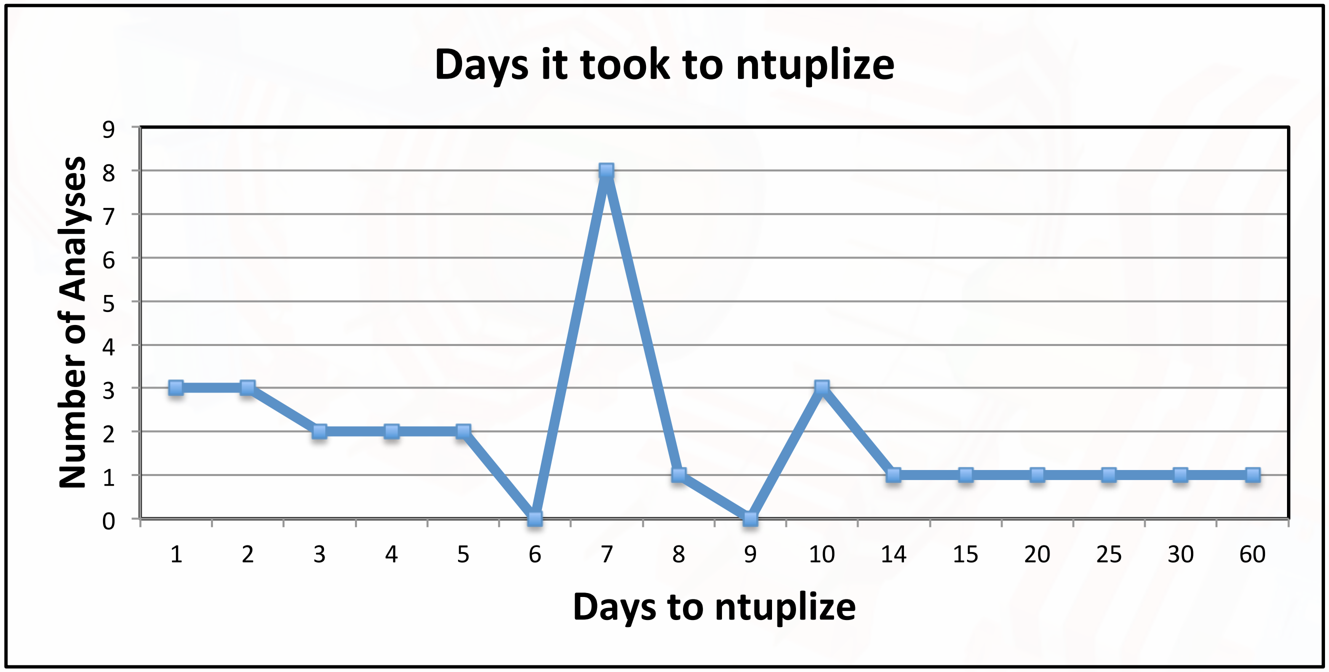
Ntuples is the popular data format for final plots for analysis



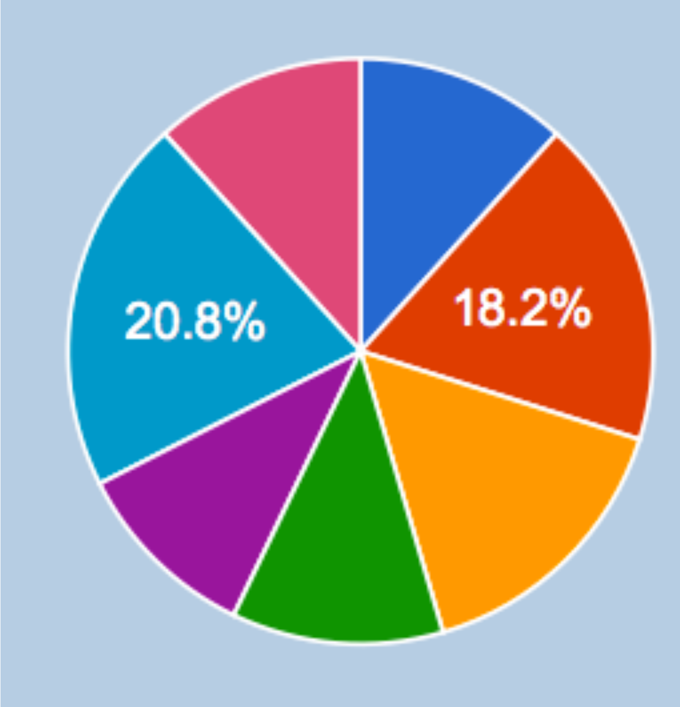


Means 4 analyses Ntuplized 10 times

How long does it take for you to n-tuplize?



Reasons to recreate ntuples



- 🟦 Change of definition of lepton isolation
- 🟥 Change of object Id
- 🟧 Change of jet calibration
- 🟩 Change of MET definition
- 🟪 Change of a high level analysis quantity
- 🟦 Change of an event pre-selection
- 🟥 Change of the definition of object cross cleaning / event interpretation

Number of grid jobs run to create ntuples

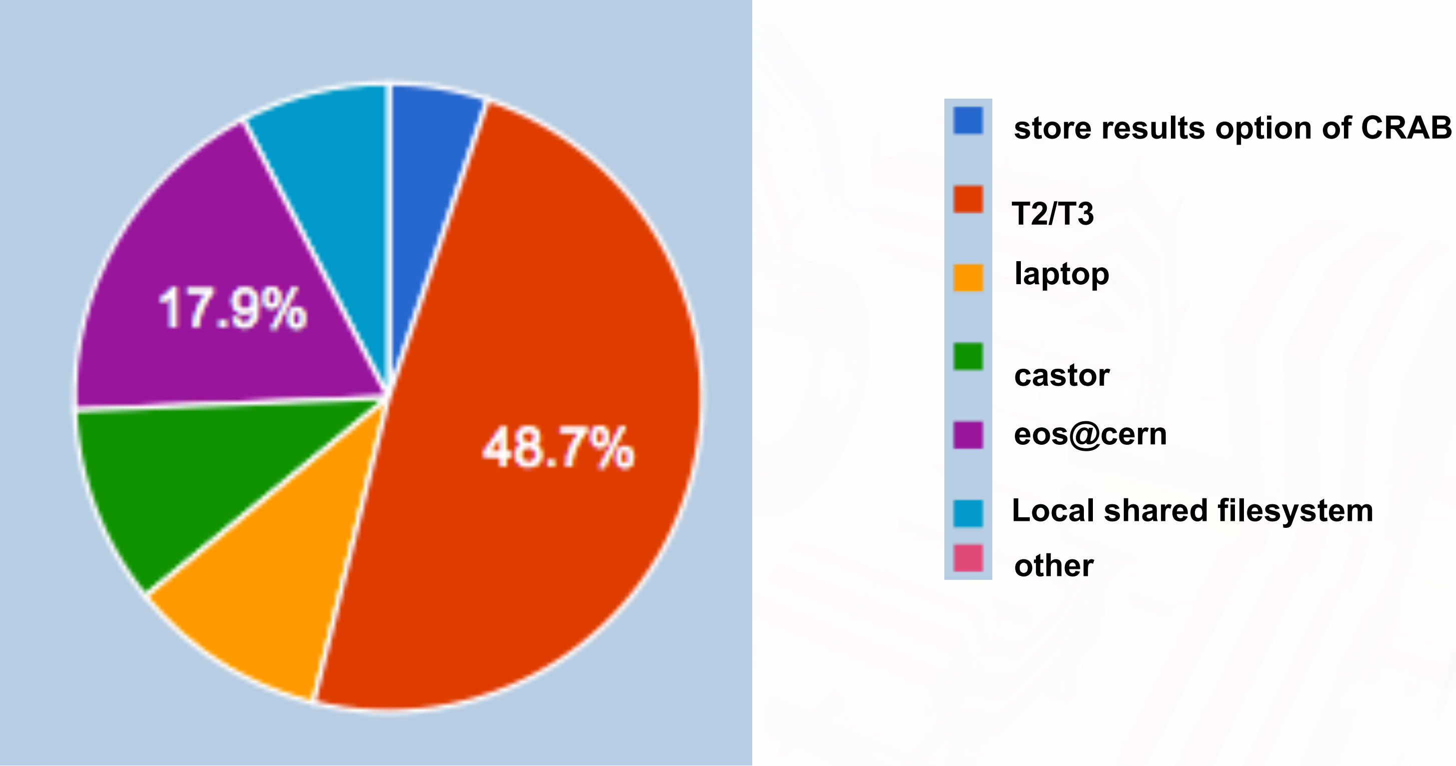
Number of grid jobs run to create ntuples

- 500 – 100K
- Some use condor batch

Typical running time for your grid jobs to produce your n-tuple (running time of a single job)

- Few hours to less than a week
- Depends on grid sites health

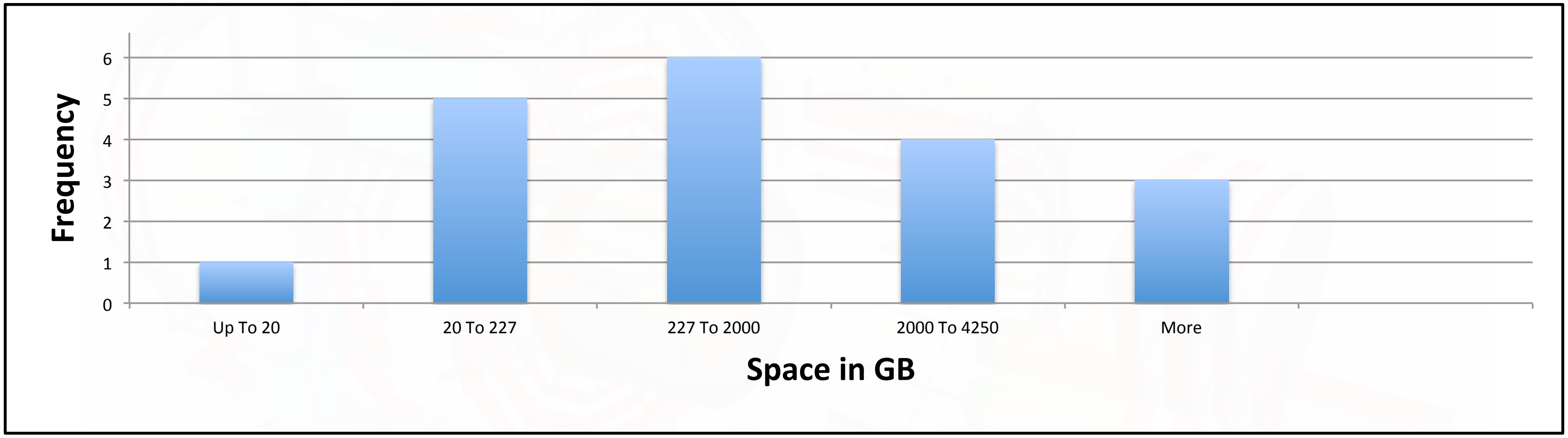
Where do you store the output of your n-tuple



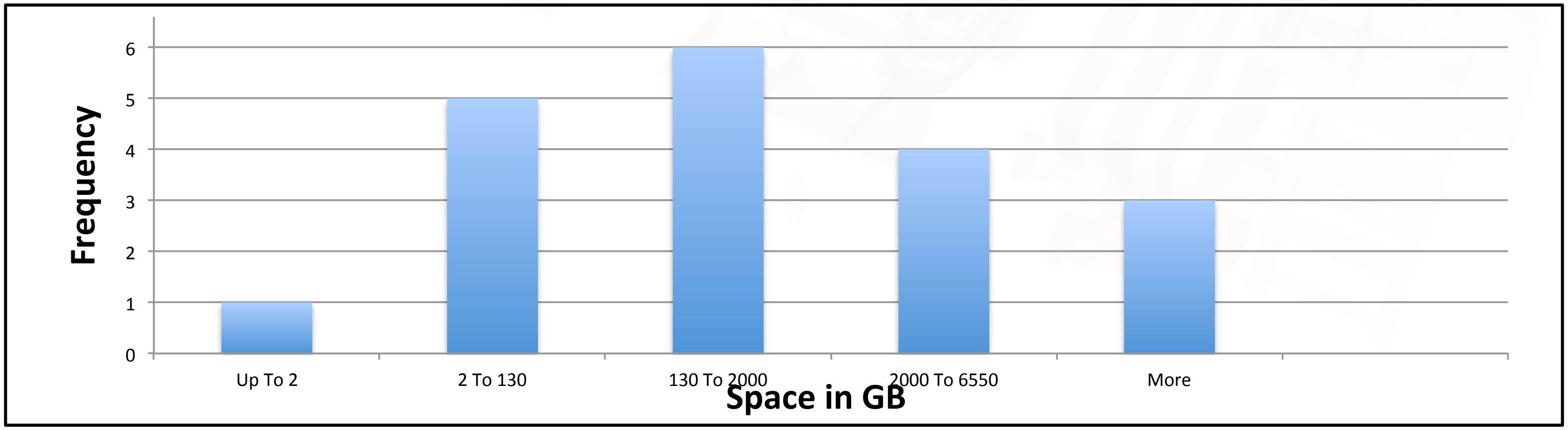
10% also store at eos@FNAL

Disk Space n-tuple (in GB)

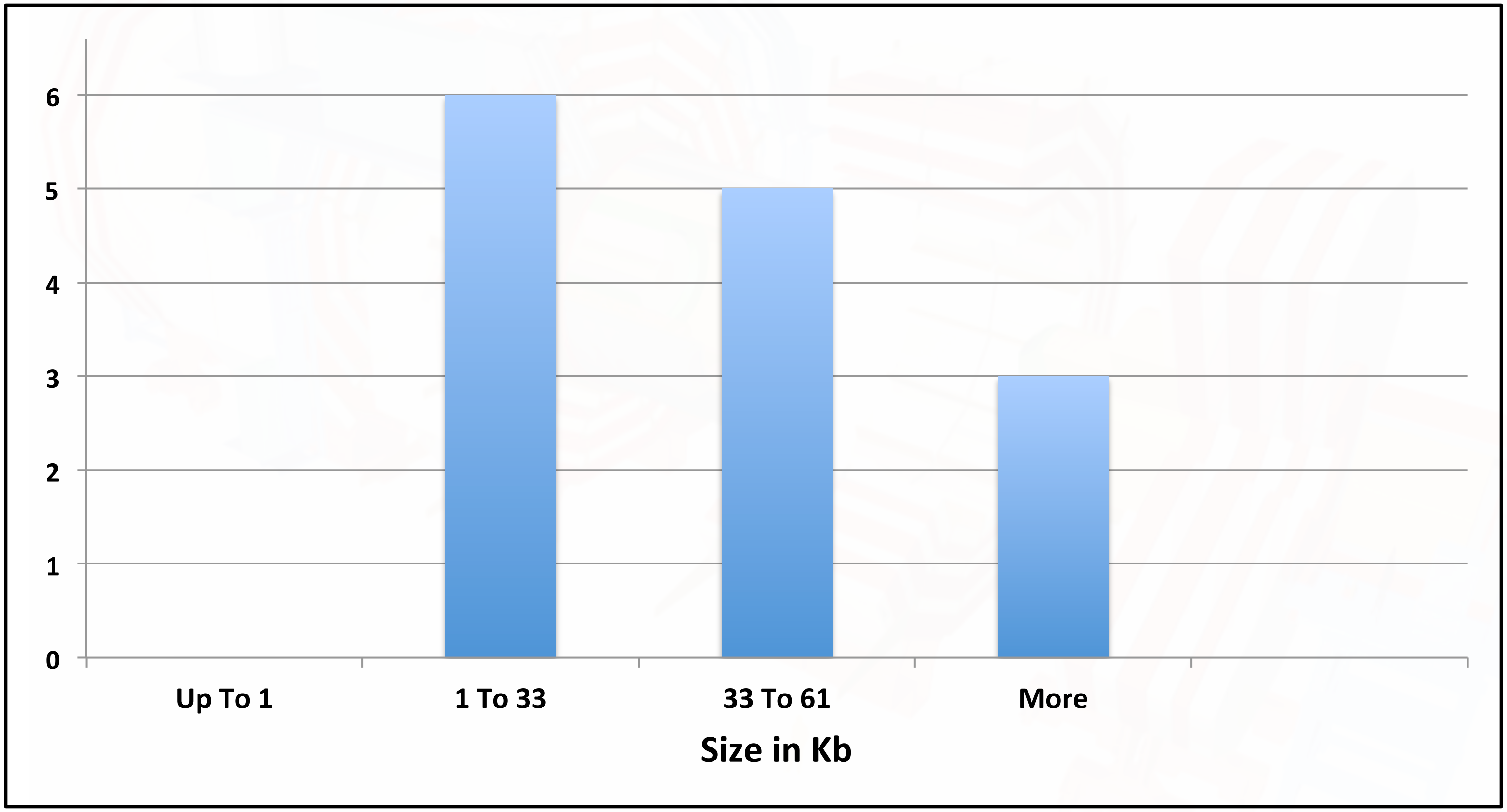
MC



Data

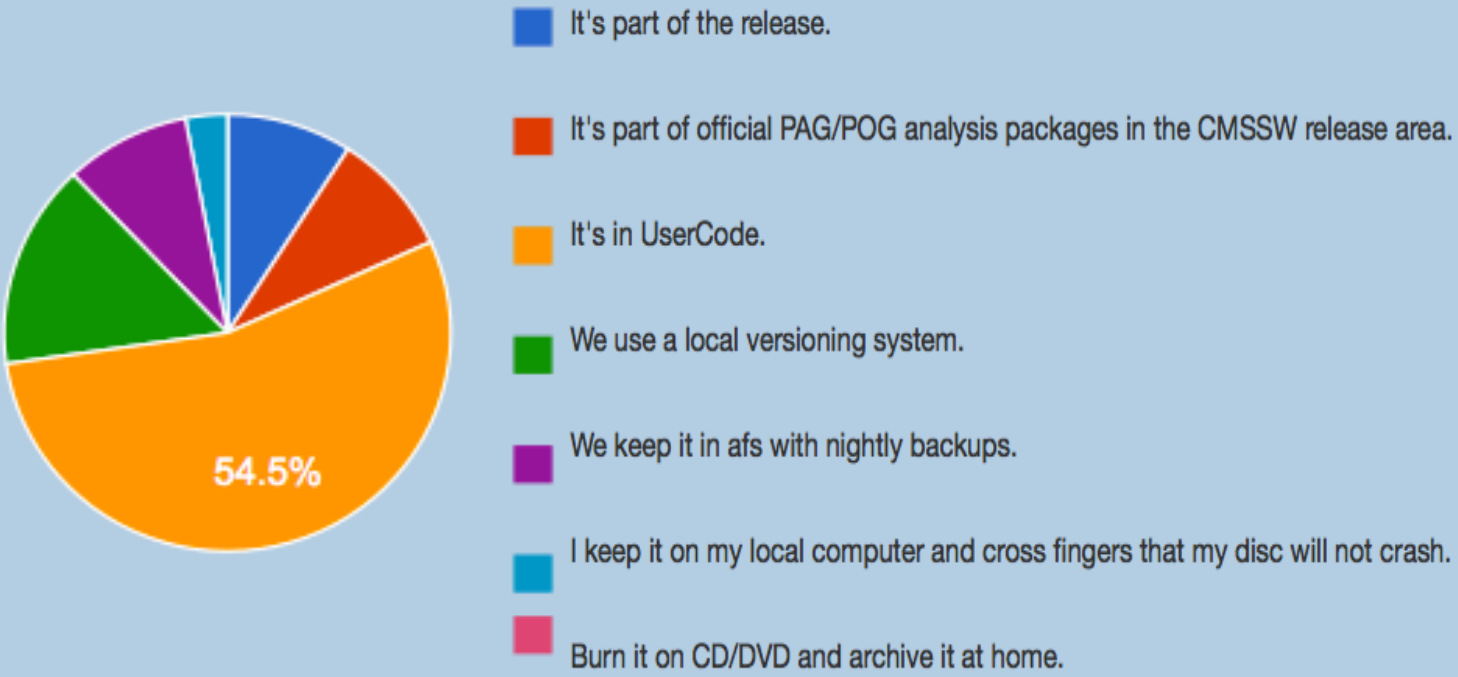


What is the size of your n-tuple per event (in KB/ event)?

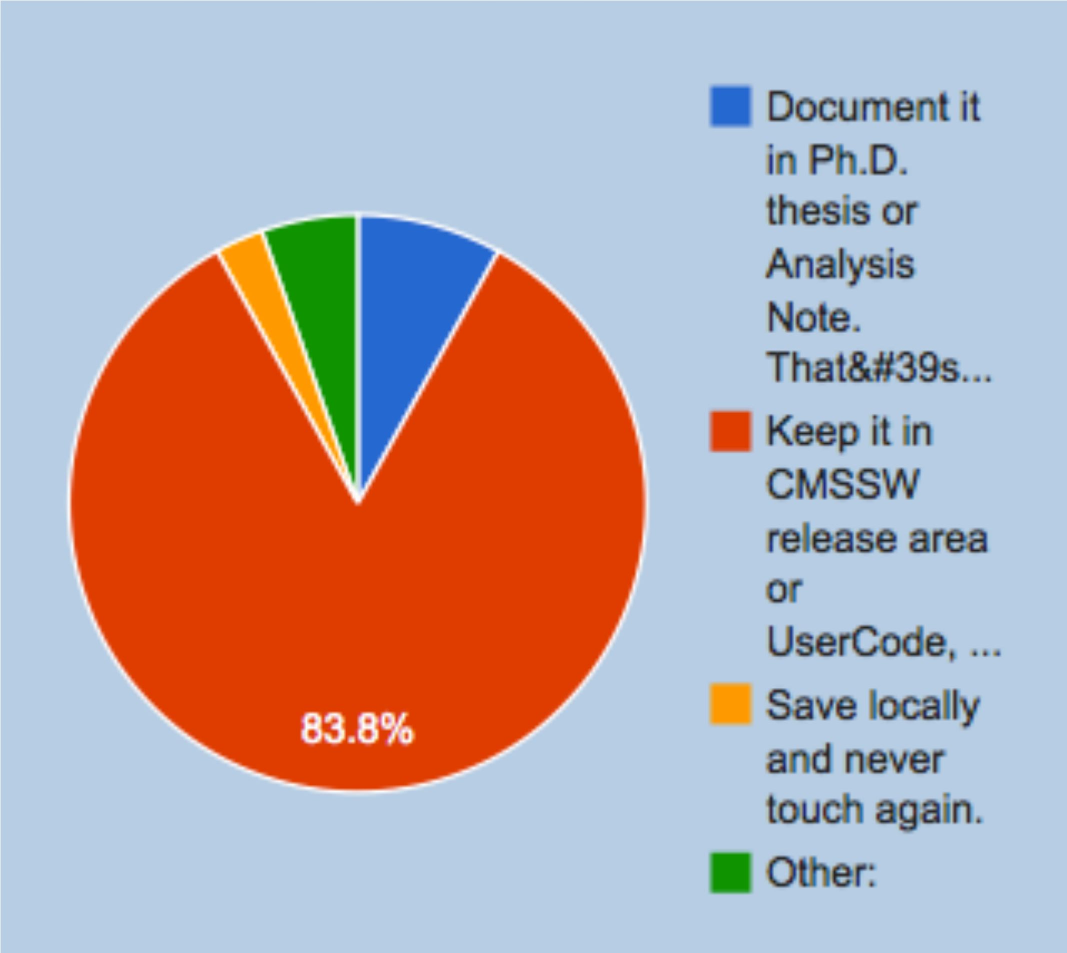


How do you foresee to preserve your analysis?

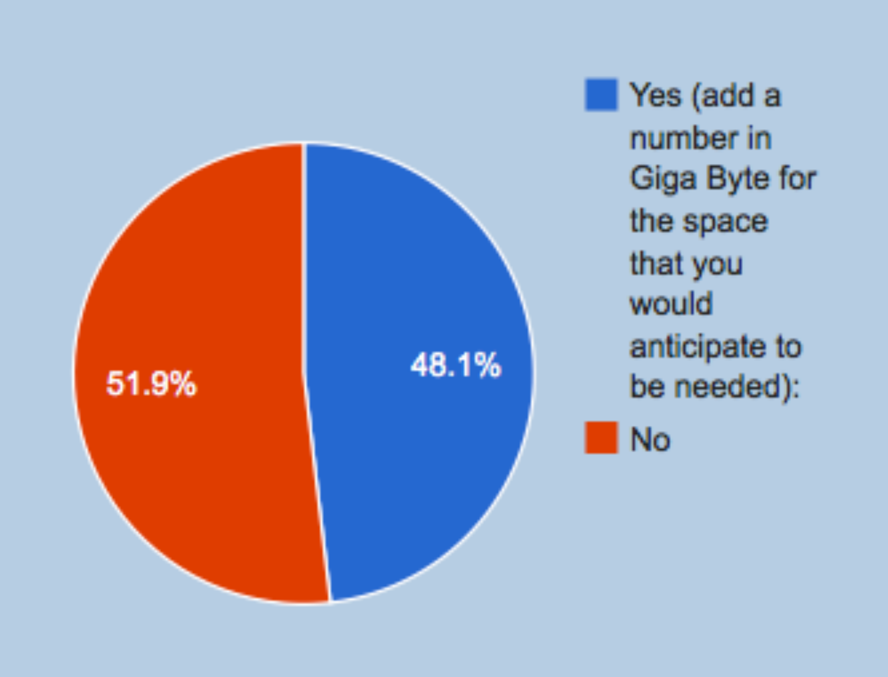
Where do you keep the code with which analysis was performed



Plan to preserve your analysis code for future?



Store ntuple for data preservation and disk space needed?



disk space needed?

80 GB to 20 TB

Summary

- CMS has a flexible software framework that covers all software needs from data taking to physics analysis
  - This is a major plus for data preservation
- CMS analysis is mostly a two step process
  - Computing intensive part on the grid
  - Final analysis, plots on local computing
- Majority analysis use PATtuples in physics analysis
- Final plots are made using ntuples and their shelf life is the duration an analysis is done
- But storing PAT tuples and AOD data gives ability to generate ntuples with modification and changes and redo variations of analysis
- Data storage is space intensive and analysis computing analysis
- Key analysis should be preserved to the extent of level 3
- Given current practices in CMS – robust documentation, CMSSW Reference Manual, Data Analysis schools etc., preserving data analysis for ourselves should be able to be implemented

Useful Links

CMS public page

- http://cms.web.cern.ch/

CMS code respository

- http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/

CMS Reference Manual

- http://cmssdt.cern.ch/SDT/doxygen//

CMS WorkBook

- https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBook/

CMS Data Analysis Schools

- https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookExercisesCMSDataAnalysisSchool