PRINCETON UNIVERSITY

Using CMS monitoring towards better models of HL-LHC (and Run3) computing

David Lange

March 12, 2019

Why do we need tools for projecting computing resource needs?

- Provide input to external reviews
  - Short-term estimates of needs (Typically CPU, disk, tape by computing tier)
  - Long-term projections for Run 3 and HL-LHC
- Understand the implications of evolutions to computing models
  - Identify critical components that drive resource needs
  - **Demonstrate impact of physics choices and R&D activities**

The idea behind models is quite simple

All the work we want to do

Map resources to activities

$$Resource_i\ (t) = \sum_{j}^{Activities} Resource_{ij}\ [t, LHC(t), HLT(t), PM(t), AM(t), TE(t), policies, \ldots]$$

Total need for resource type i

The idea behind models is quite simple

$$Resource_i\ (t) = \sum_{j}^{Activities} Resource_{ij}\ [t, LHC(t), HLT(t), PM(t), AM(t), TE(t), policies, \ldots]$$

All the work we want to do

Map resources to activities

Trigger rates into offline

Analysis model

Total need for resource type i

Machine plans + performance

Production model

Technology evolution

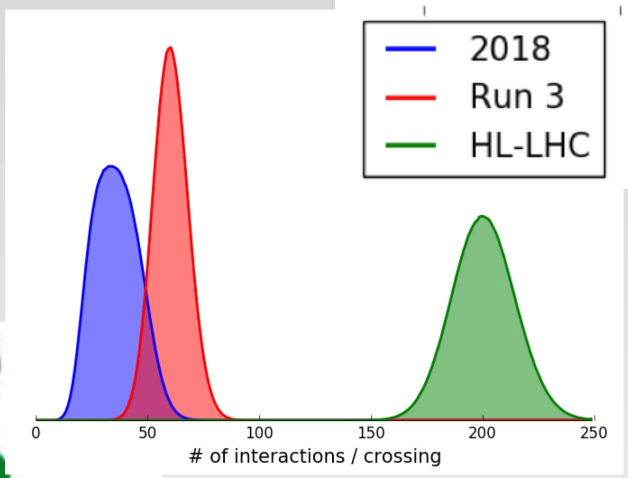
The idea behind models is quite simple

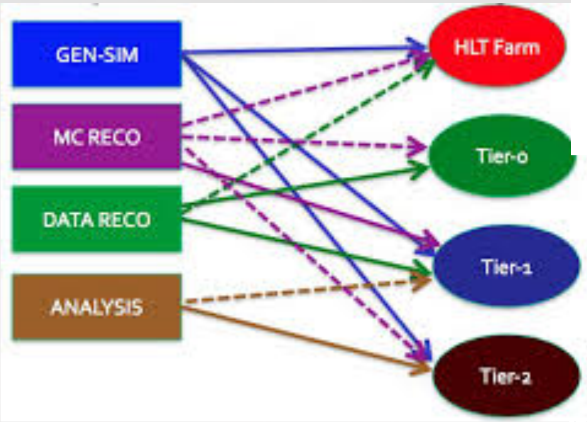
$$Resource_i\ (t) = \sum_{j}^{Activities} Resource_{ij}\ [t, LHC(t), HLT(t), PM(t), AM(t), TE(t), policies, \ldots]$$

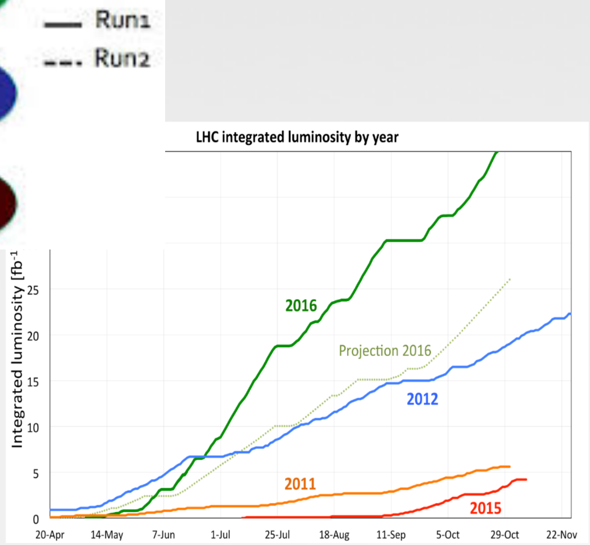
- Total need for resource type i
- All the work we want to do
- Map resources to activities
- Machine plans + performance
- Trigger rates into offline
- Production model
- Analysis model
- Technology evolution
- Resource management

This complexity comes from having a truly complex environment and the need to model it in detail

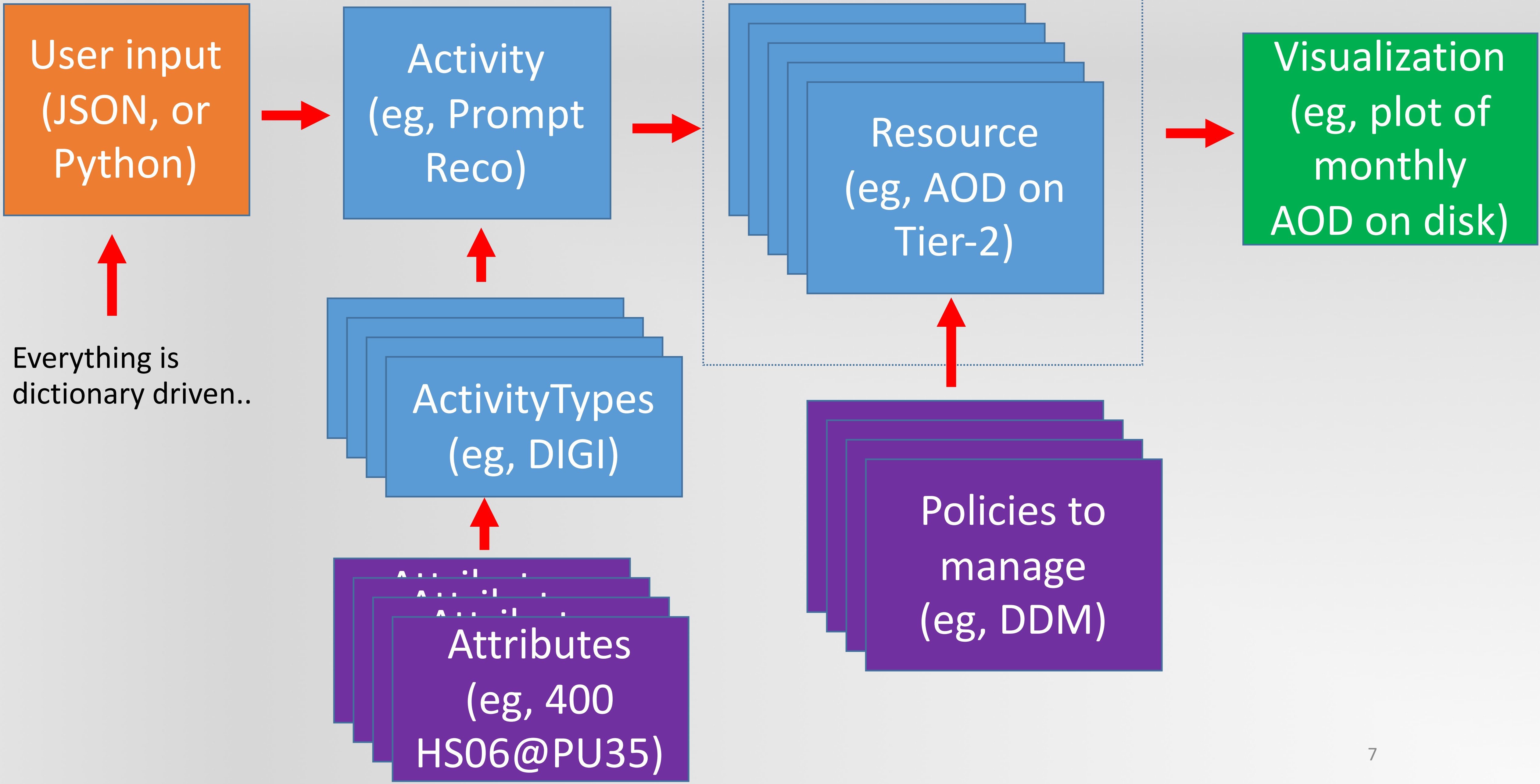
- What workflows to run? Where? and when?
- Evolution of experiment workflows, data tiers, analysis requirements, etc
- Evolution with instantaneous luminosity
- Evolution with integrated luminosity
- Impact of LHC reliability
- Expected analysis user behavior
- Evolving balance of commissioning needs vs analysis needs
- Impact of site infrastructure needs
- Use of dynamic and heterogeneous resources
- Policies that ensure efficient resource usage (e.g., data management policies)







Current Approach



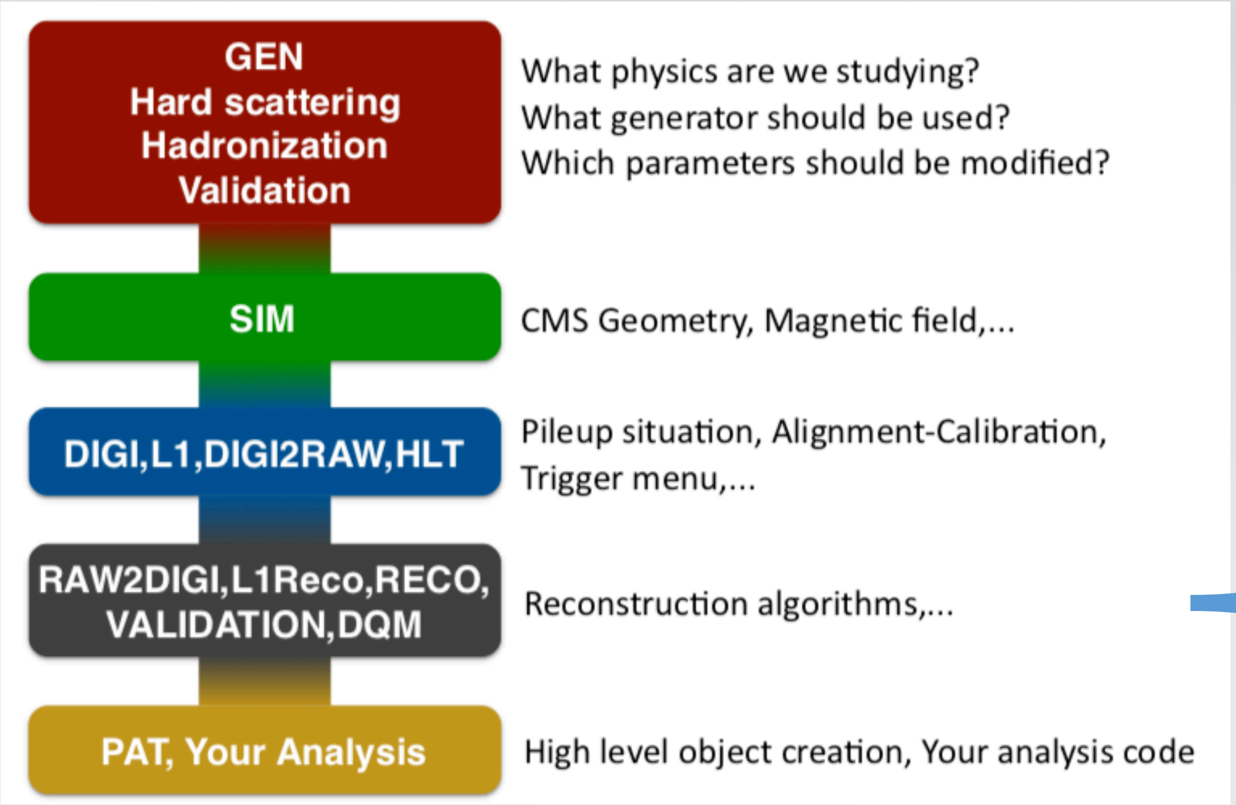
How do we figure out the values and dependencies of all of these parameters?

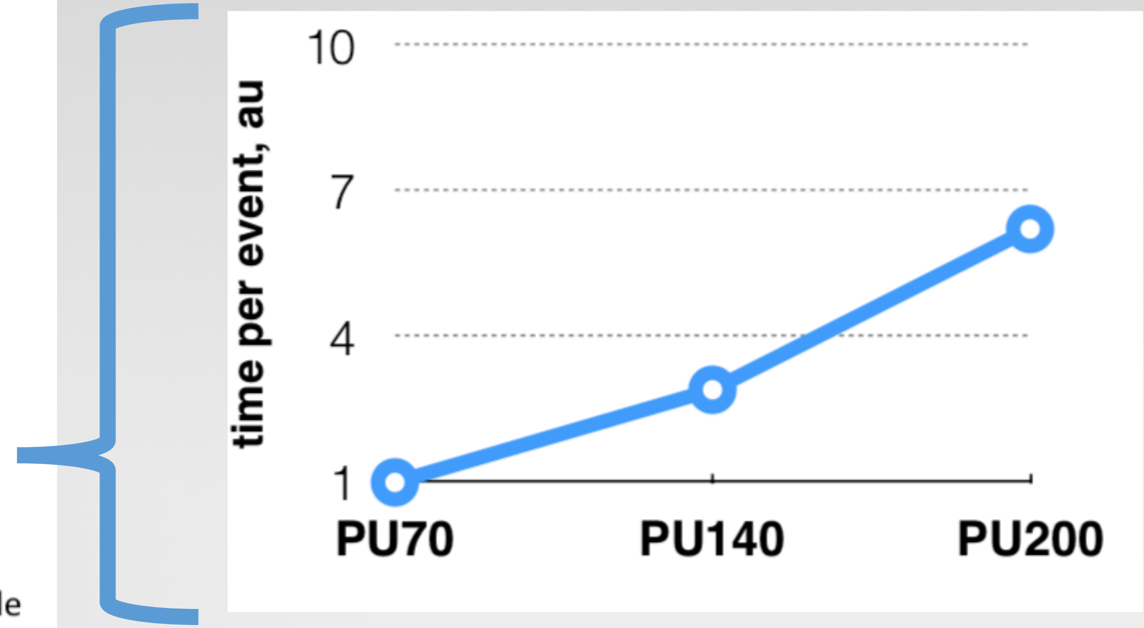
Begin with current practices build up model of computing activities over time

- Define production and analysis activities in terms of their:
  - Resource needs (CPU, storage, network, etc)
  - Input event rates (Dependent on delivered luminosity, LHC livetime, etc)
  - Begin/end schedule (Eg, Need-by dates)


```
{ "label" : "PROMPTRECO",
  "attributes" : [ { "type" : "Tier-0 CPU",      "purpose" :    "RECO CPU",     "fractionOfEvents" : 1.0   },
                   { "type" : "Tier-1 Disk",     "purpose" :    "RECO storage", "fractionOfEvents" : 0.1   },
                   { "type" : "Tier-0 Disk",     "purpose" :     "AOD storage", "fractionOfEvents" : 0.125 },
                   { "type" : "Tier-1 Disk",     "purpose" :     "AOD storage", "fractionOfEvents" : 0.25  },
                   { "type" : "Tier-2 Disk",     "purpose" :     "AOD storage", "fractionOfEvents" : 0.63  },
                   { "type" : "Tier-1 Disk",     "purpose" : "MINIAOD storage", "fractionOfEvents" : 1.0   },
                   { "type" : "Tier-2 Disk",     "purpose" : "MINIAOD storage", "fractionOfEvents" : 3.0   },
                   { "type" : "Tier-1 Tape",     "purpose" :     "AOD storage", "fractionOfEvents" : 1.0   },
                   { "type" : "Tier-1 Tape",     "purpose" : "MINIAOD storage", "fractionOfEvents" : 1.0   },
                   { "type" : "Tier-1 Tape I/O", "purpose" :     "AOD storage", "fractionOfEvents" : 1.0   },
                   { "type" : "Tier-1 Tape I/O", "purpose" : "MINIAOD storage", "fractionOfEvents" : 1.0   },
                 ]
}
```

Current application performance parameters are big factors





- These are straightforward to estimate (well, except for analysis processing)
  - Measure locally (Typically strongly dependent on pileup level)
  - Factor in realities of running in production environments (realistic CPU efficiencies, etc)

Similar issues modeling needed data storage replicas

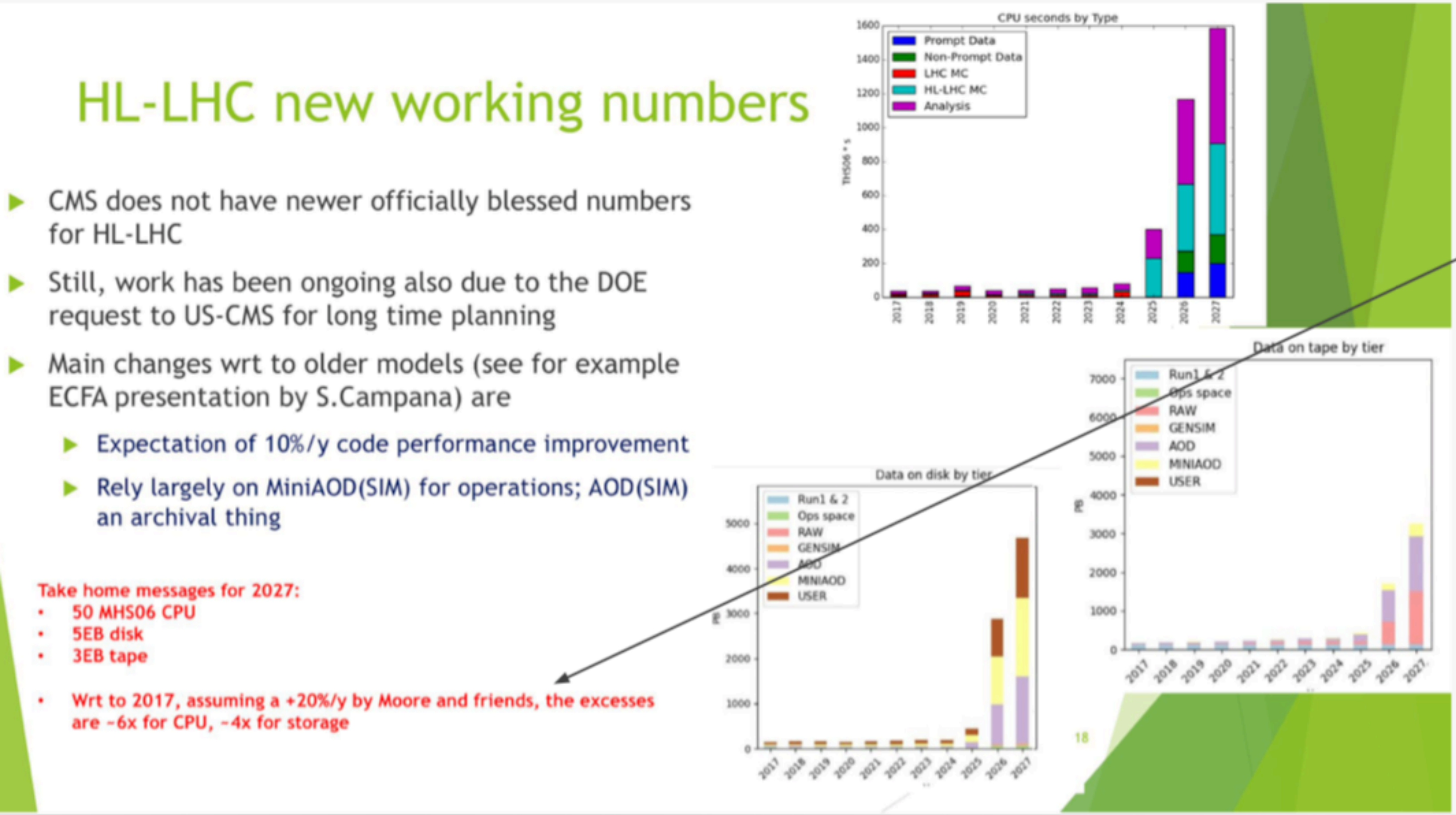
- Far more data produced than can be kept on disk.
- “Popularity” based system sets replication factors

Data tier	Run 2 rates	Disk presence	Access pattern
RAW	1 kHz * 1 MB	Minimal	Detector/trigger studies only
AOD	1 kHz * 0.4 MB	Partial (yes, if used)	Re-derivation of MiniAOD, Some analysis
MiniAOD	1 kHz * 0.04 MB	Multiple (if used)	Current the primary analysis tier
NanoAOD	1 kHz * 0.002 MB	Multiple (if used)	New analysis tier serving ~50%
Skims, Calibration samples	Varies	Yes, if used	Detector studies, Analysis

- Challenges projecting forward
  - Forecasting “quality of service” vs resource needs. Judging impact of more/less replication, impact of site metrics and variability
  - Scale of MC needs. Scaling up with luminosity
  - Further optimization of data tiers

How will we evolve into HL-LHC?

HL-LHC projections suggest that changes in the CMS production and analysis models are needed

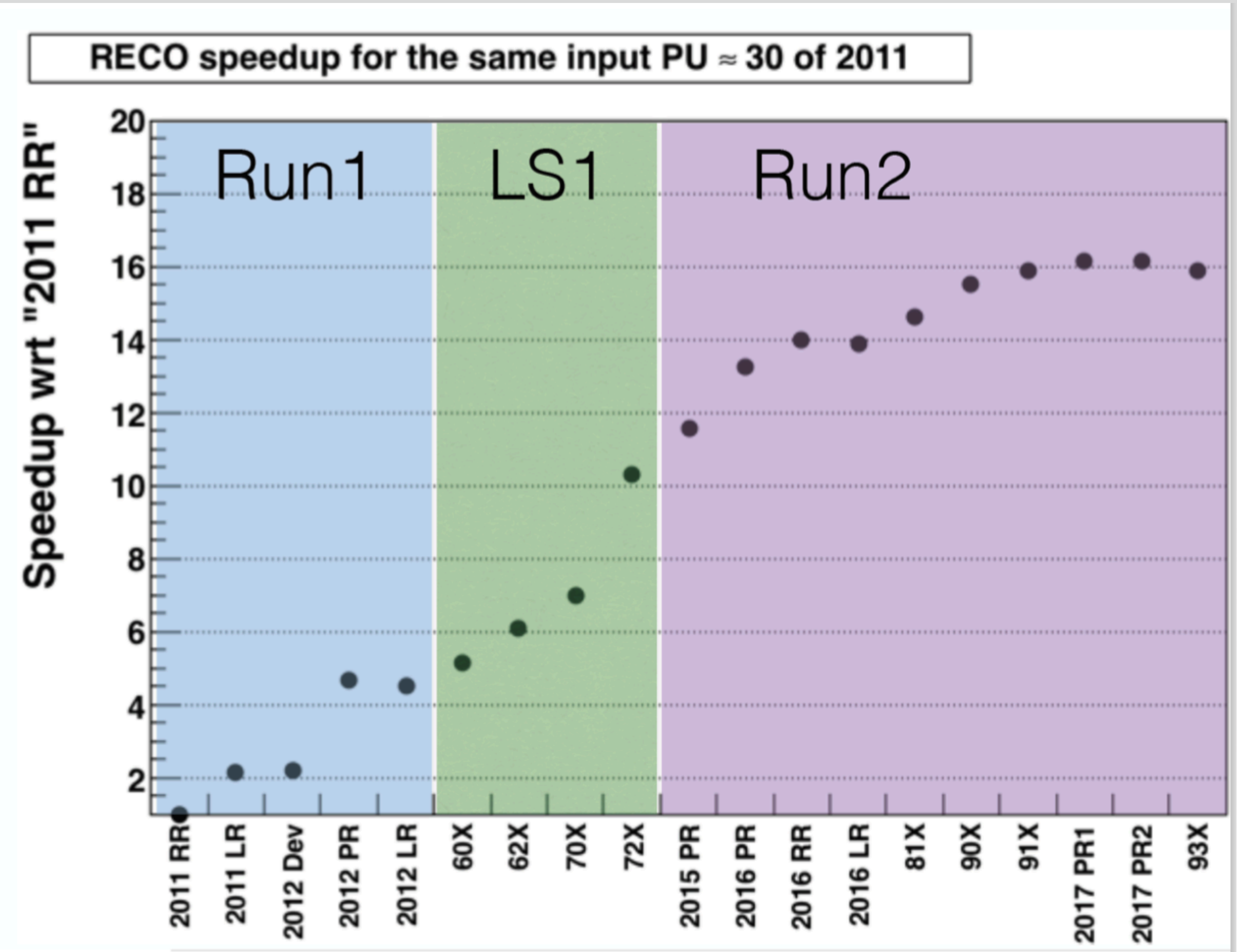


- Last "official" numbers by CMS on HL-LHC needs (Sept 2017, LHCC) are
- 4-6x excess is ON TOP of the technological improvement!
- (in other words, if we decrease the needs by 4-6x wrt these figures we would be at flat budget)
- **We have new (reduced) estimates for Sept 2018 LHCC - will be shown in one of the first meetings**

- We started a small group to look at how CMS uses its data today to motivate, evaluate and justify changes to our data model for the future

Application parameters are difficult to forecast

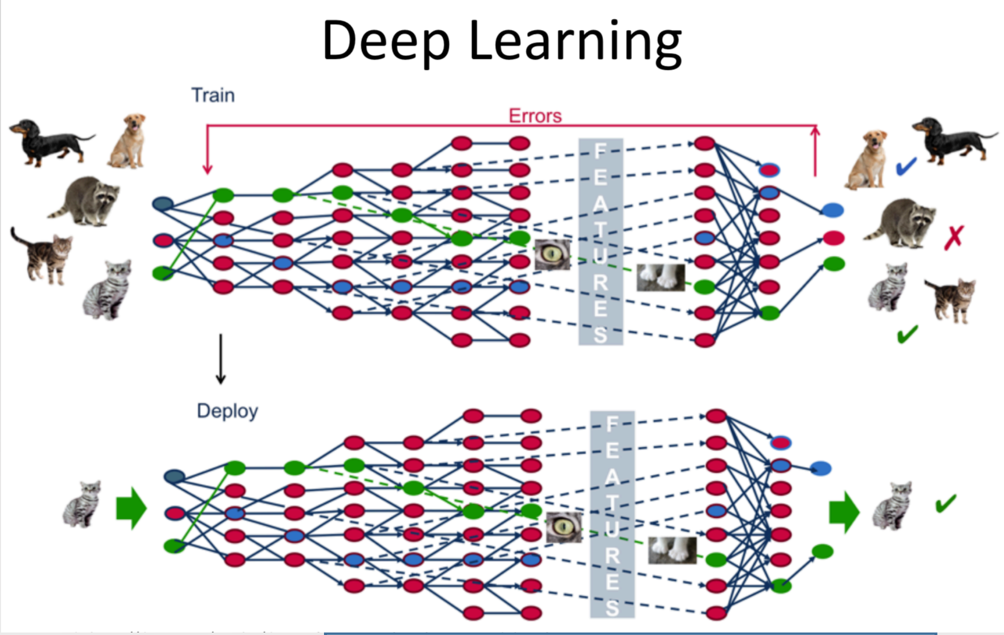
- Is past performance (improvement) a good indictor of the future?
  - Optimistically, yes, but….
  - Much of the search for gains from "**low hanging fruit**" has instead become **algorithm reengineering**
- Modeling perspective:
  - Make a projection (e.g., 10%/yr improvement in application run time)
  - Remember that the error on it enters **exponentially!**

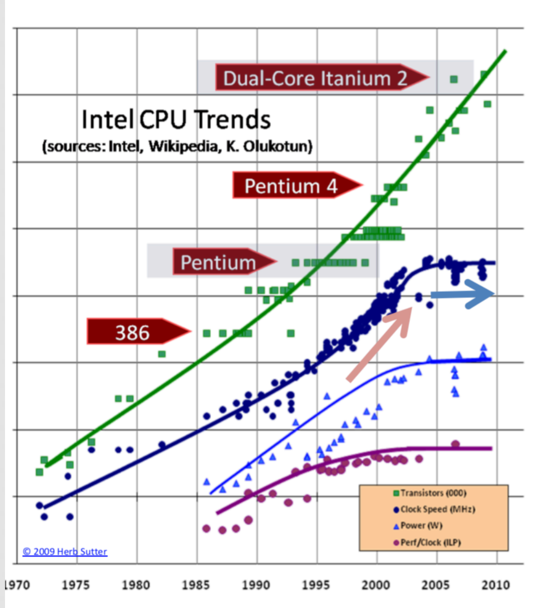


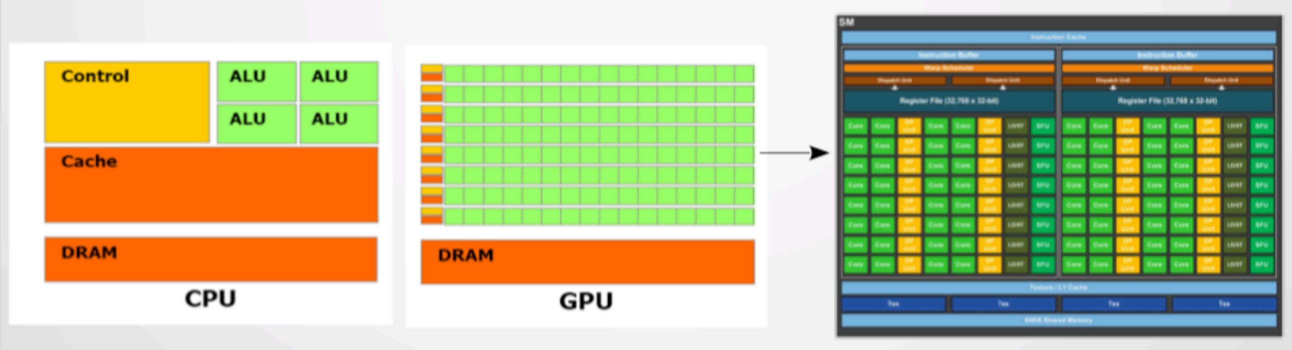
PR is prompt reconstruction
RR is (end-of-year) rereconstruction
LR is legacy reconstruction
releases ending in X are development

And then there are potential game changers

- Some of today's topical examples:
  1. Multicore and manycore architectures
  2. Accelerators (on HPCs and otherwise)
  3. Deep Learning techniques for HEP
  4. Transformative analysis model changes. Eg, the CMS NanoAOD
- Difficult to "plan" for in resource projections







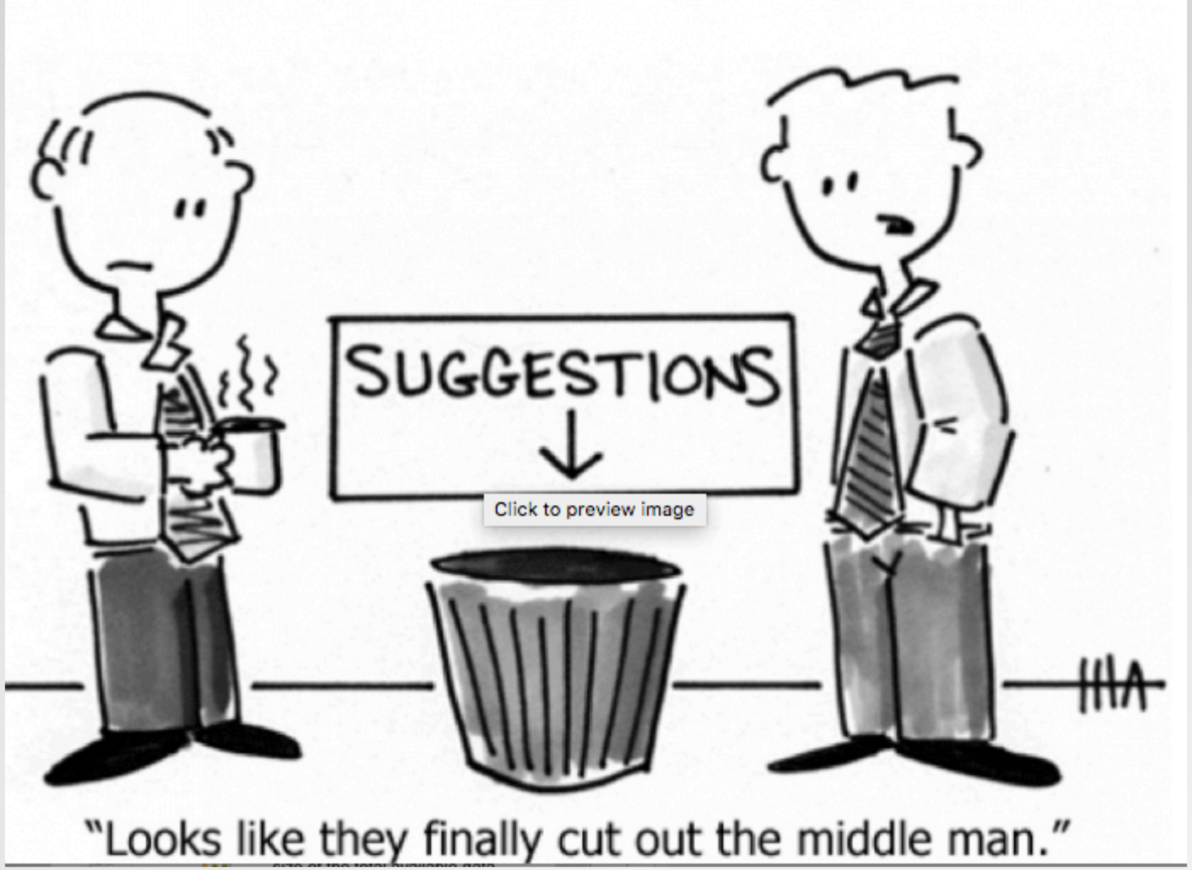
Nvidia Tesla P100 Streaming Multiprocessor (SM) architecture

CMS Data tiers in Run-2

Data tier	Size (kB/ev)
RAW	1000
AOD	400 [x8 reduction]
miniAOD	50 [x8 reduction]
nanoAOD	1 [x50 reduction]

Its easy to make suggestions

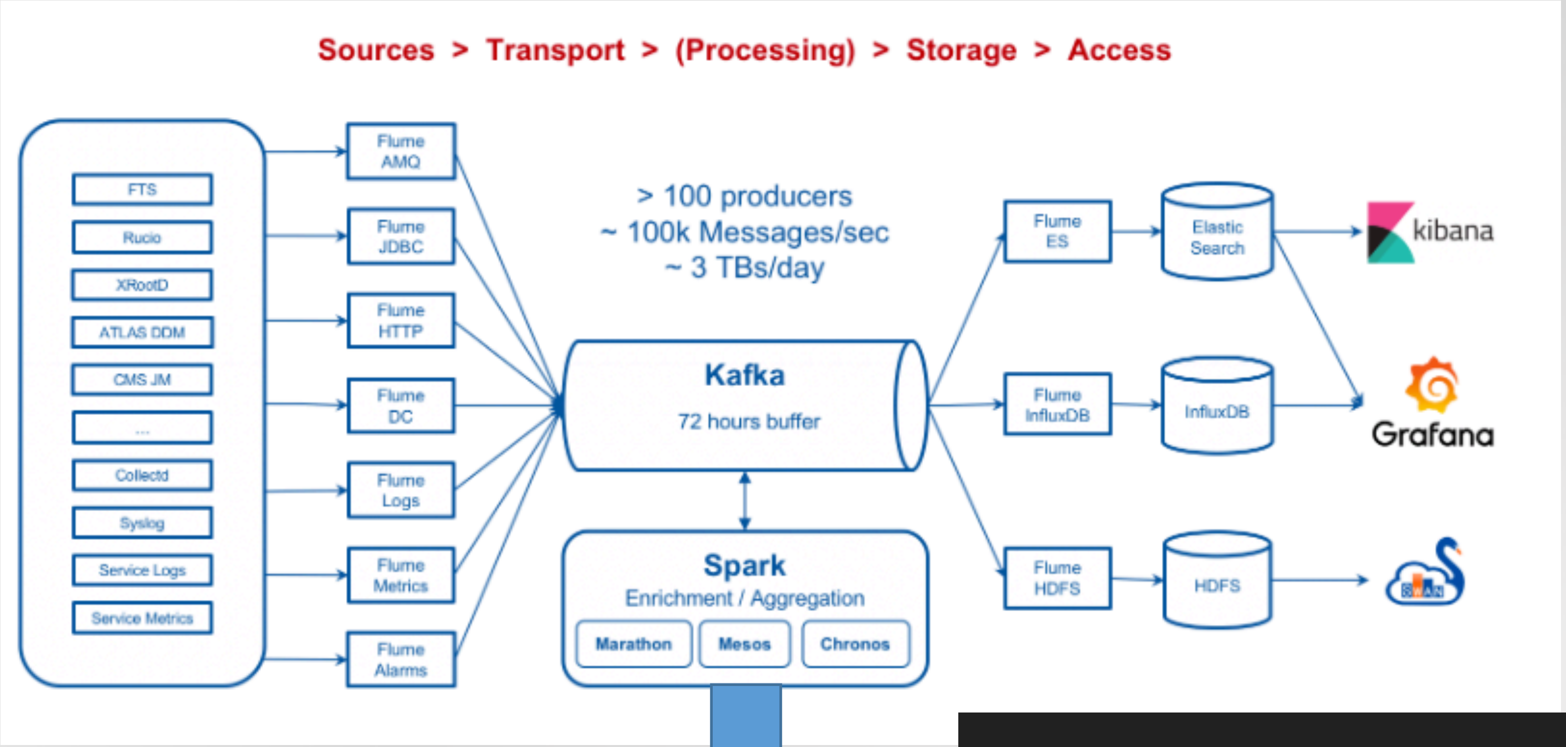
- Not-so prompt reconstruction
- More MC for ML
- Data Caching
- Fast(er) MC
- Analysis Facilities
- HPC
- Data lakes
- Real-time analysis
- GPUs
- Deep learning

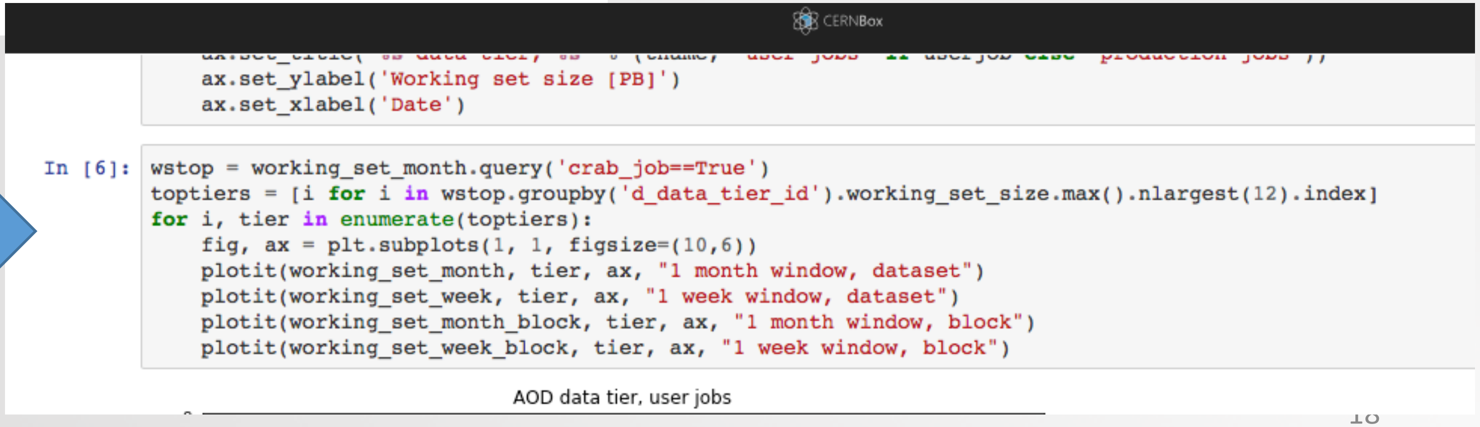


Models backed up by reality are an excellent evaluation tool for planning computing model evolution in the "right" way

Motivations to support these ideas as being realistic are more difficult

Our approach – leverage available data sources monitoring user and production jobs and data availability





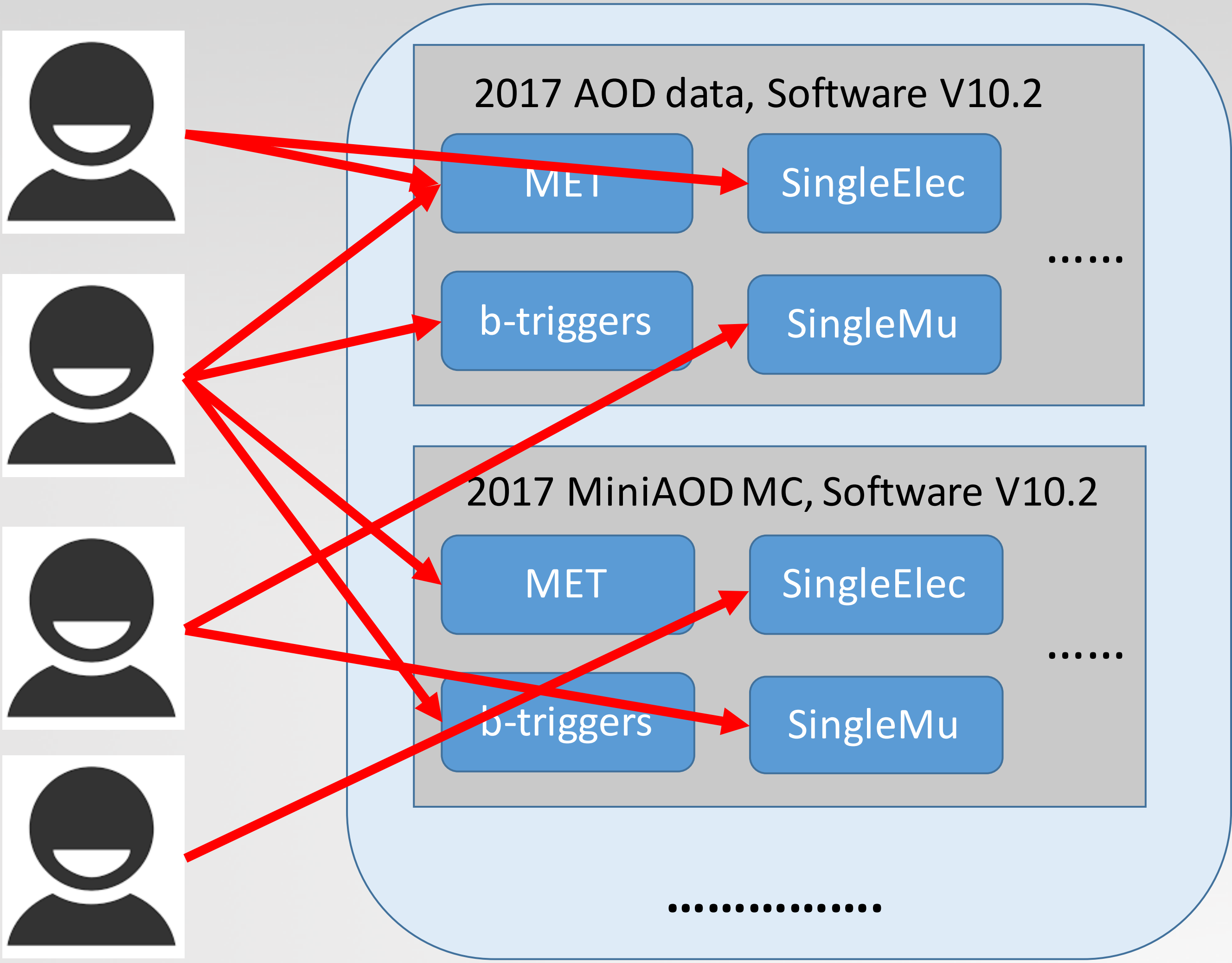
How do analysts actually use CMS data (modulo holes in our monitoring data…)

- Monitoring tells us
  - How often data sets are read – **On average very often**
    - Great, we have data that is interesting to our experiment
  - How much CPU is consumed per event – **Not very much**
    - This makes sense – most processing is done by central production. Only particle identification and similar recipes remain for analysts to perform (typically)
  - How this CPU has increased with PU – **Not very much**
    - This make sense – the big combinatorical algorithms are running in the event reconstruction not in analysis
  - What data is used – **Almost entirely the ”new” but not “newest data”** – eg, 2016 data sample was the dominant sample used during 2017
    - Lower luminosity or lower energy data less topical
    - Current years data is still being understood by experts

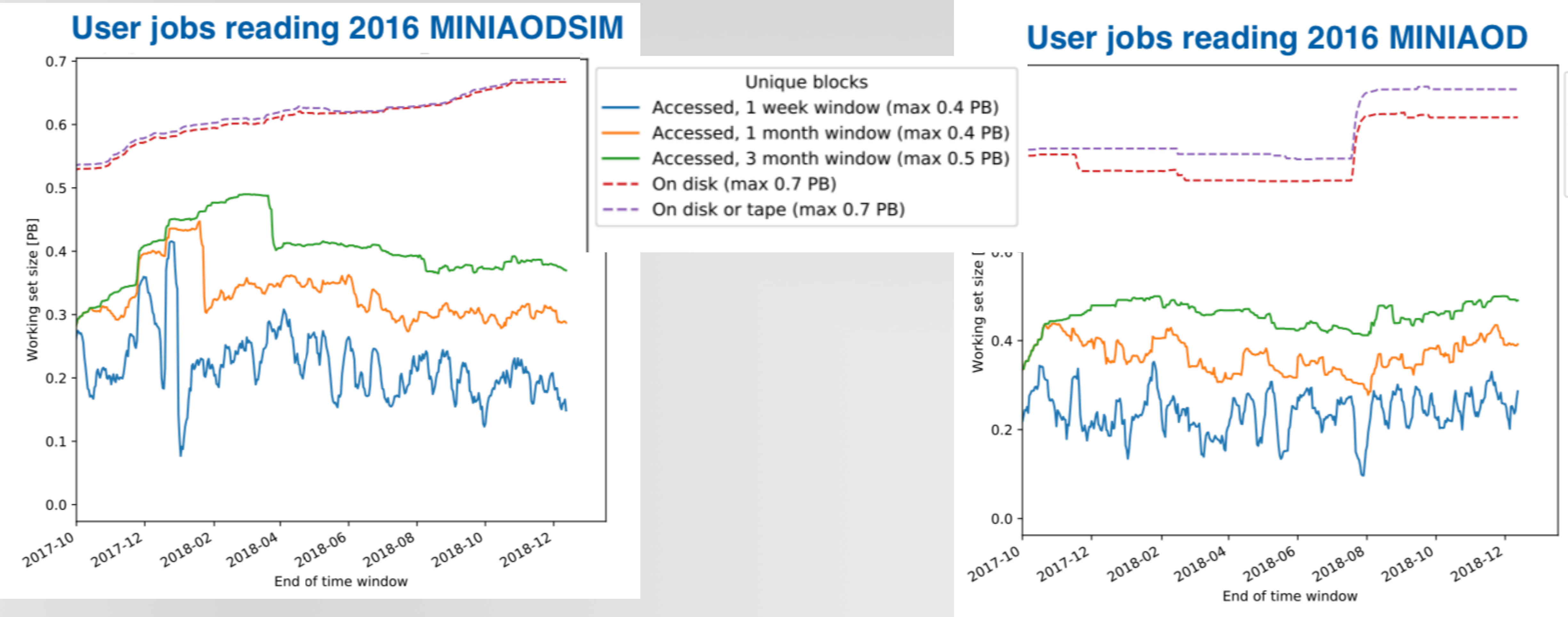
This motivates the concept of an "working set" of CMS

- We define the "working set" as the sum of all datasets needed by analysts in a given time window (eg, during 3 months)

- Needs to assess the CMS working set, its characteristics and its evolution
    1. Attributes of **datasets and files** (eg dataset size, data taking era, data tier)
    2. **Subscriptions of dataset replicas** to CMS Tier1 and Tier2 sites (eg, historical start and end date)
    3. Production and analysis **file accesses** (file opens, events read)



Example results on evolution of analysis usage of 2016 MINIAOD



What do we learn? (so far)

- The sum total of the most popular data/MC in the most popular data tier for CMS is less than 1 PB

- 30% - 50% of data/MC read each week

- The 3 month working set is twice the size of the one week working set
  - Much of the data accessed this week will be accessed again soon

- Many open questions still to be solved. For example - can we predict the data sets of next weeks working set?
  - Some parts are easy: MC events for the large background samples done with the latest generators and corresponding to the popular data taking period
  - But can an automated algorithm do better? Needed to keep tape recall rates in balance

Conclusion

- Working towards a computing model backed up by understanding the behavior of analysts and our production system

- Future projections will get better
  - Modify parameters to reflect current behavior learned from monitoring
  - Extendable models are important for capturing more complexity and including observables beyond requirements on HS06 of CPU and PBs of storage.
  - Capturing correlations is a must for understanding the entire system. Solutions may solve one bottleneck by creating another

- We (will) have a good basis for evaluating concepts for the future
  - Cost per service a good, if hard to quantify, metric.
  - At minimum, better judge the effects of suggestions on demands/requirements of evolving technology.