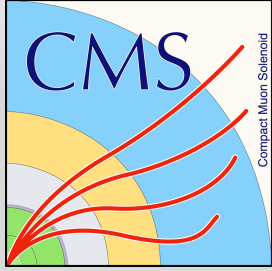


Modeling of the CMS HL-LHC computing system

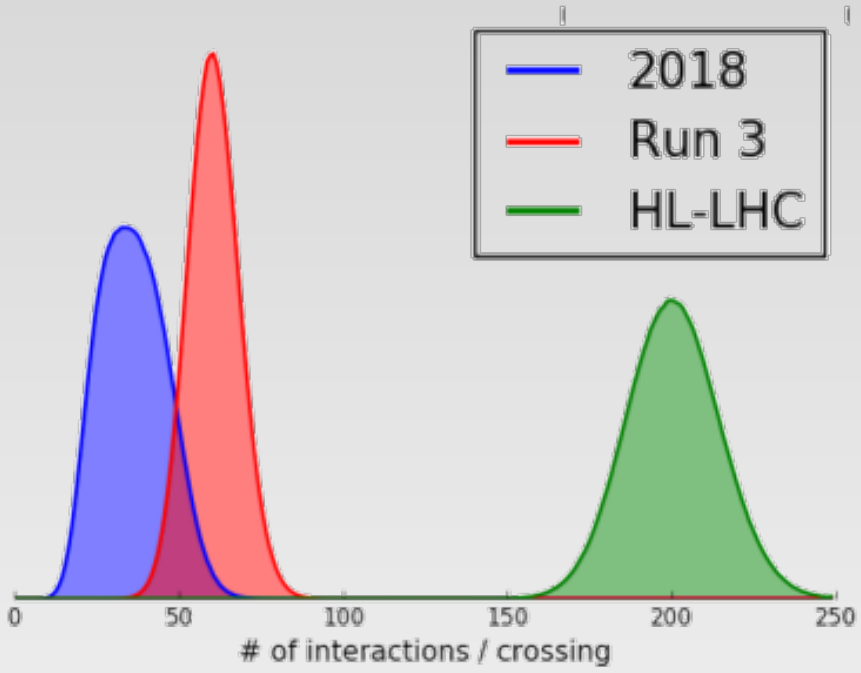
David Lange

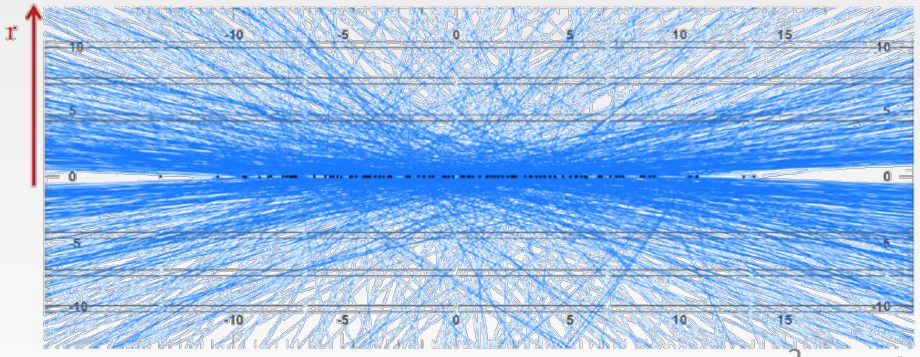
Princeton University

November 4, 2019

Physics challenges to CMS software/computingduring HL-LHC

- Increase luminosity means more interactions per bunch crossing. This brings a number of new challenges:
  1. **Higher detector occupancy**: More sophisticated detector technologies and higher channel count
  2. **Trigger**: Higher rates needed to preserve current physics reach; Use capabilities earlier in the processing chain (e.g., tracking at level-1 trigger) and real-time analysis concepts
  3. **Particle reconstruction**: More difficult to separate patterns means physics impact (eg efficiency vs fake tradeoff) and technical performance (e.g., CPU time) challenges
  4. **Analysis sensitivity**: Searches for lower cross section processes demand higher precision and most robust reconstructed data

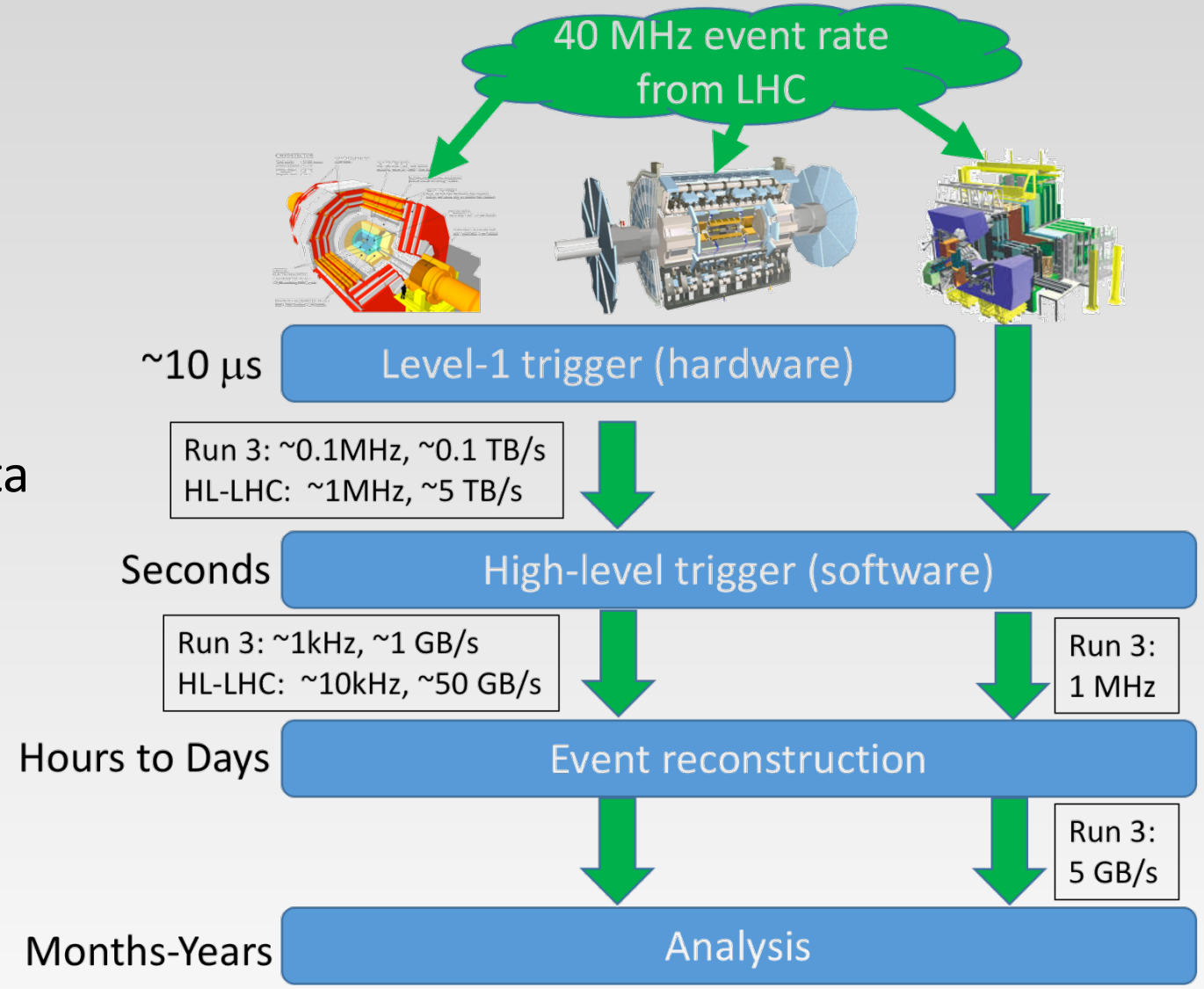




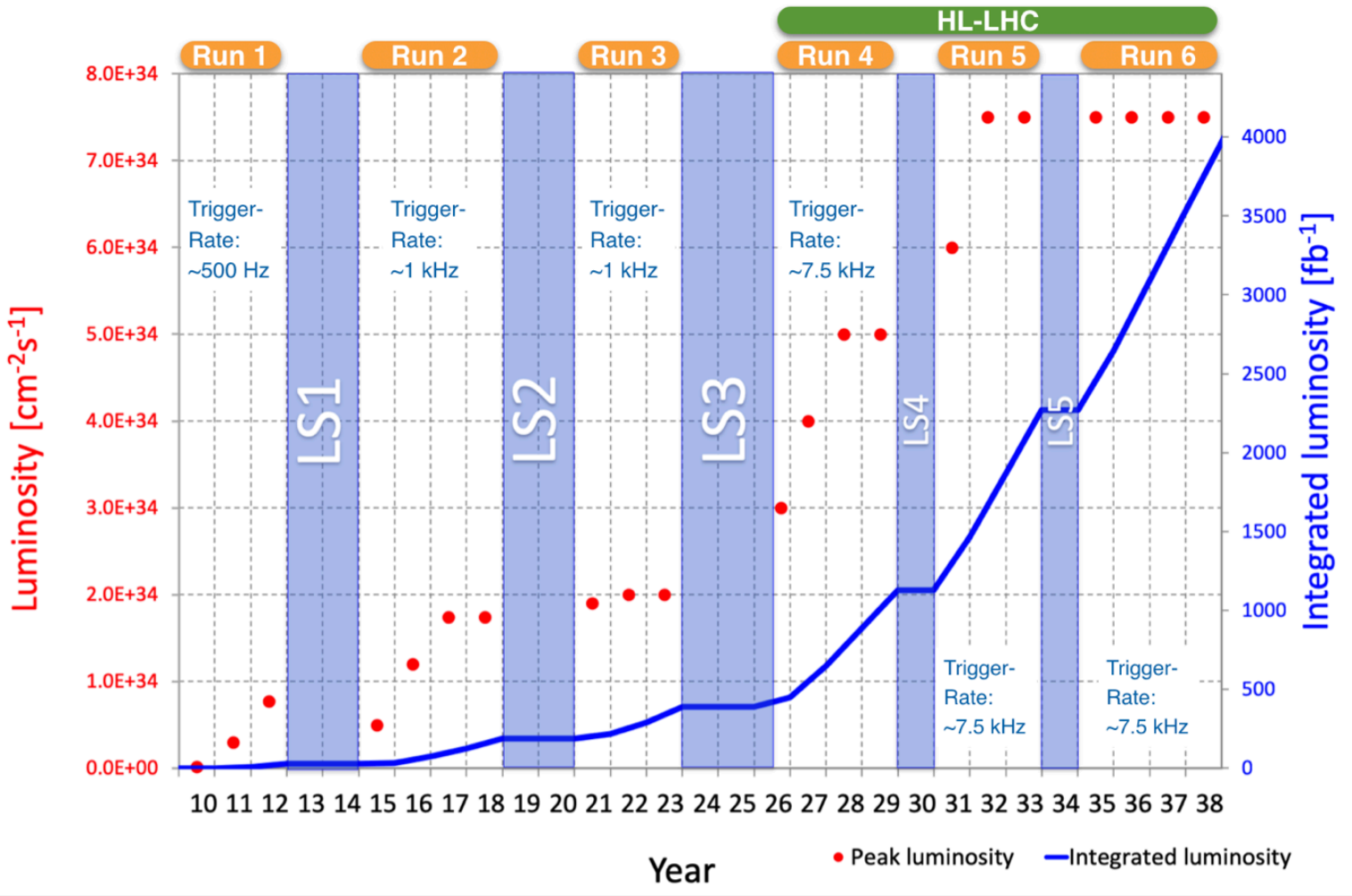
Exascale CMS driven by science

- CMS is looking at how to evolve current practices to handle much larger data rates in the HL-LHC era
  - O(20 EB/yr) into HLT
  - O(0.2 EB/yr) saved RAW data
  - O(0.1 EB/yr) analysis data and simulation samples

We use models of resource needs to estimate computing needs (and costs) as well as R&D impacts



The HL-LHC operations schedule extends out 20 years



Models that estimate computing resource needs (and costs) can answer a number of important questions:

- How will the computing need for the experiment evolve over time?
- How would operational changes affect the computing need for the experiment?
- What is the impact on on-going R&D or potential R&D on resource needs?
- How does technology evolution impact operational models and resource needs?

Ideally, computing should never delay the science....can we assure resource providers that this is true into the future?

The resource modeling approach in CMS

- Estimate from bottoms up based on
  - LHC operations and luminosity profile
  - Needs defined by physics drivers (e.g., needed trigger rate and data formats to carry out the analysis program)
  - Build rough timeline of activities (e.g., when and where)
  - Measurements of resource “cost” of each activity (e.g., how much CPU, disk, tape, network…)
  - Estimates of future improvements

- This technique started as a spreadsheet based model for short term and has developed into a Python tool capable of short and long term projections

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

All the work we want to do

Map resources to activities

Trigger rates into offline

Analysis model

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

Total need for resource type i

Machine plans + performance

Production model

Technology evolution

The idea behind models is quite simple

All the work we want to do

Map resources to activities

Trigger rates into offline

Analysis model

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

Total need for resource type i

Machine plans + performance

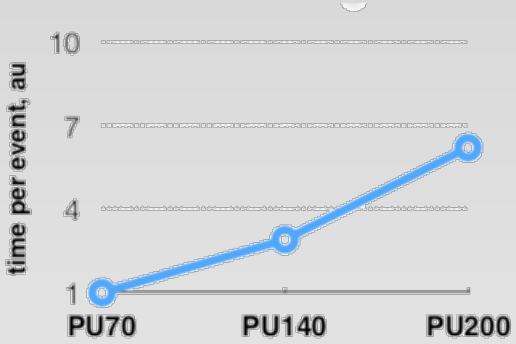
Production model

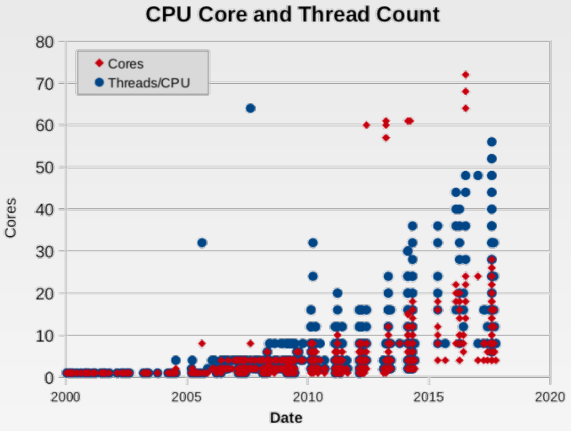
Technology evolution

Resource management

Gathering input parameters from a number of sources

- LHC planning:
  - Run schedule
  - Operational expectations for luminosity / efficiency
- CMS detector and physics programs:
  - Needed data tiers to perform timely analysis
  - Needed event rates to record physics program
- CMS and external software development teams
  - Application performance as a function of LHC conditions and planned evolution
  - Data storage requirements by data tier as a function of LHC conditions and planned evolution
- Computing hardware costs and evolution
  - Hardware is often hidden behind HS06 and PB until cost is included (eg, $/HS06, $/PB)
  - Heterogeneous systems (eg hardware accelerators) will change this..



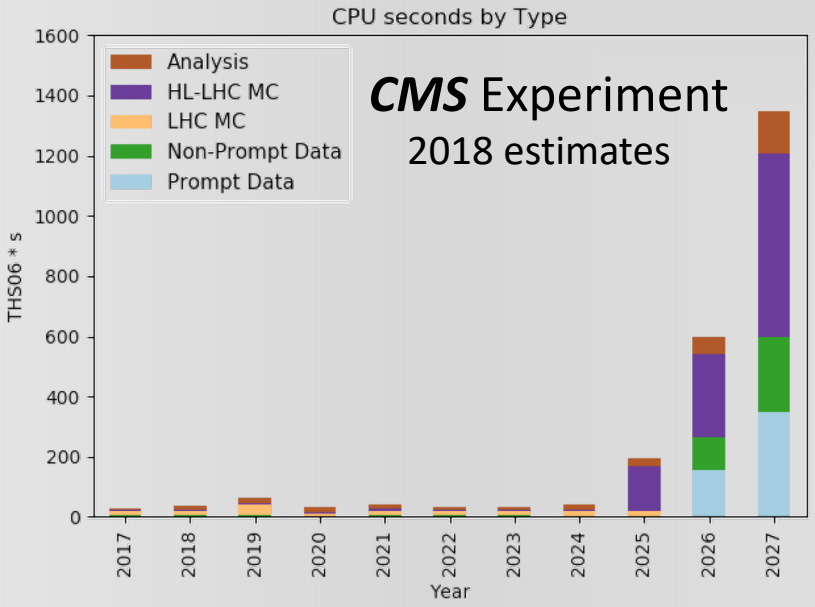


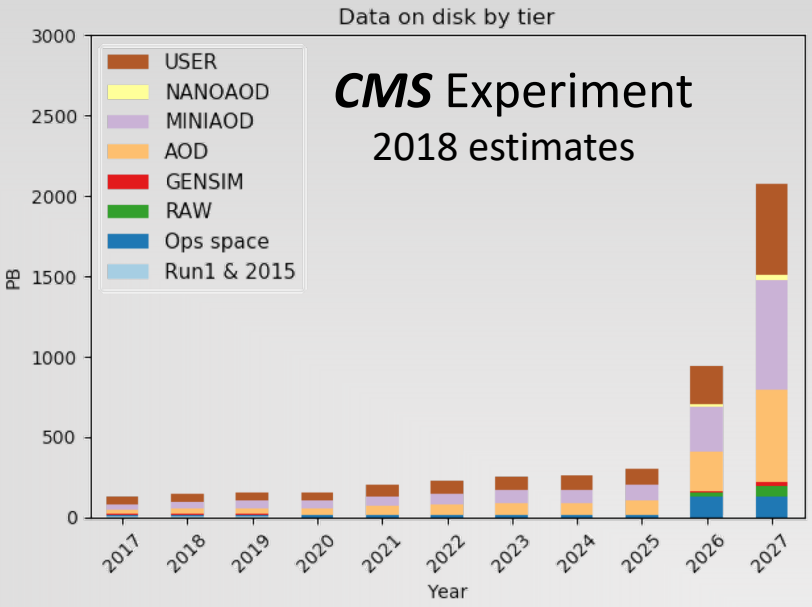
CMS Data tiers in Run-2

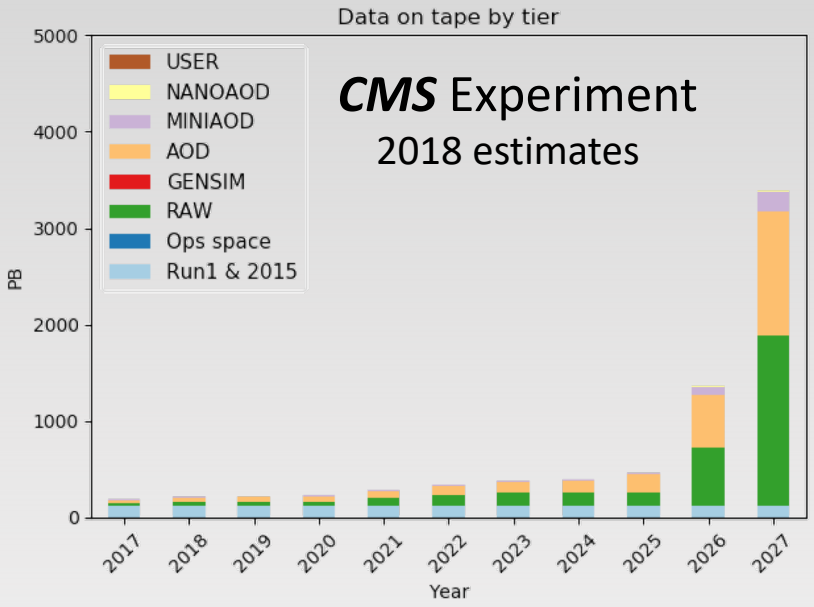
Data tier	Size (kB/ev)	
RAW	1000	
GEN	< 50	
SIM	1000	
DIGI	3000	
RECO	3000	
AOD	400	[x8 reduction]
miniAOD	50	[x8 reduction]
nanoAOD	1	[x50 reduction]

Analysis formats

From these inputs, we derive resource need estimates



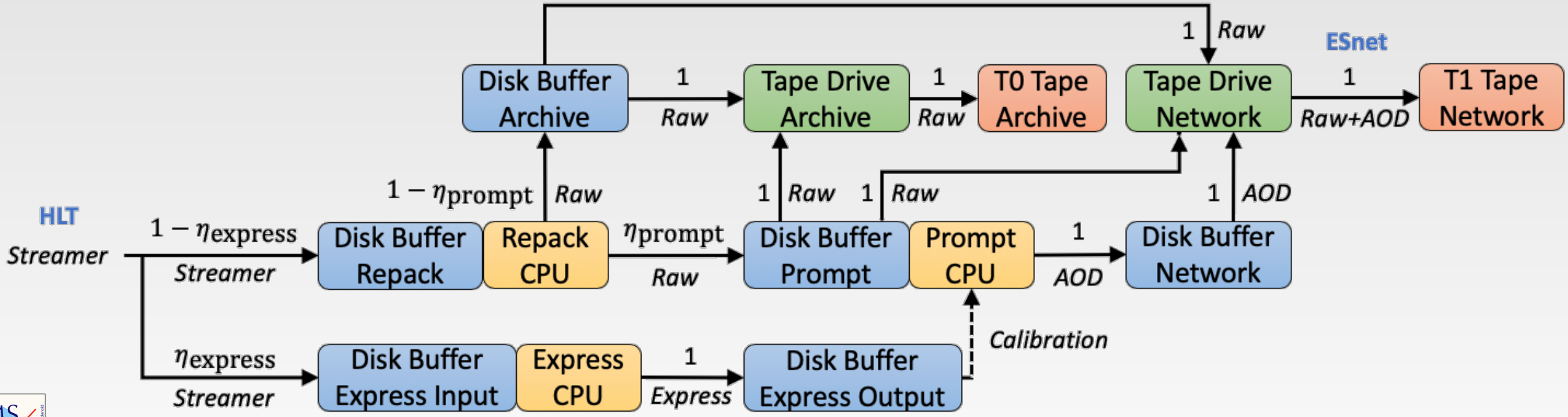




- Update on requirements from CMS is coming soon. We are making refinements via on-going ECOM process in CMS which is looking at ideas for evolving the computing model for HL-LHC (and is aiming to finish this year)

An alternative approach – system cost modeling

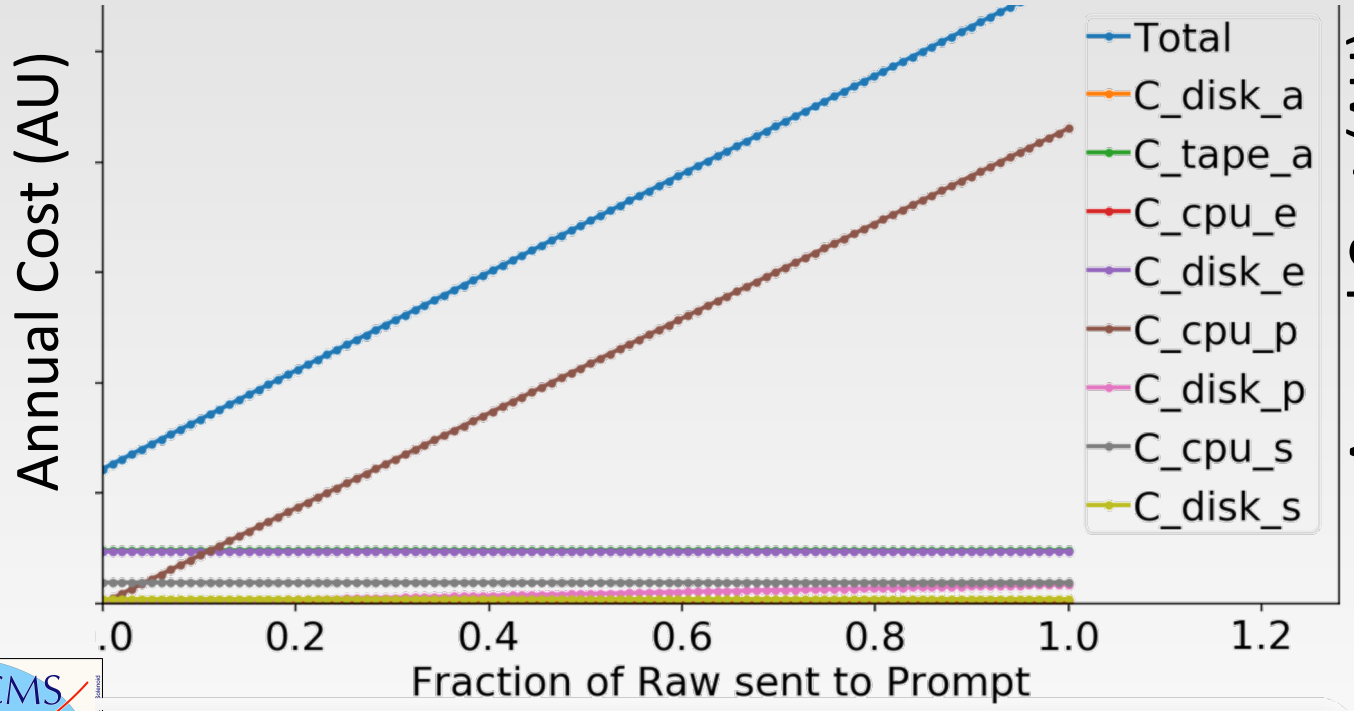
- Attempt to build detailed model of system components (eg, the Tier-0 facility), then look at cost tradeoffs.
- For example a notational (and incomplete) model of the CMS Tier-0 workflow
  - Based on similar input parameters as with an overall resource model, but with the added ability to evaluate the impact of more detailed operational changes



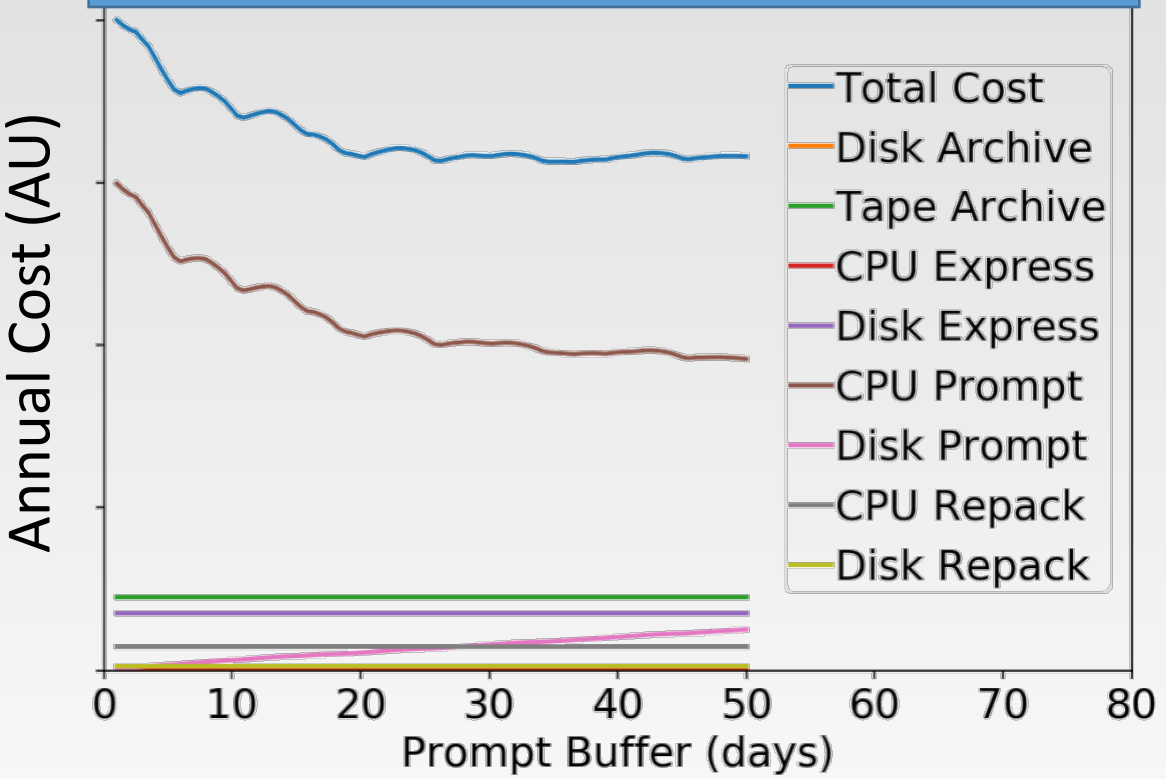
This approach facilities the evaluation of cost tradeoffs vs operational practices and understanding which things are cost drivers

- This is very much a work in progress, but some notational examples to illustrate the idea

Fraction of data going to prompt reco

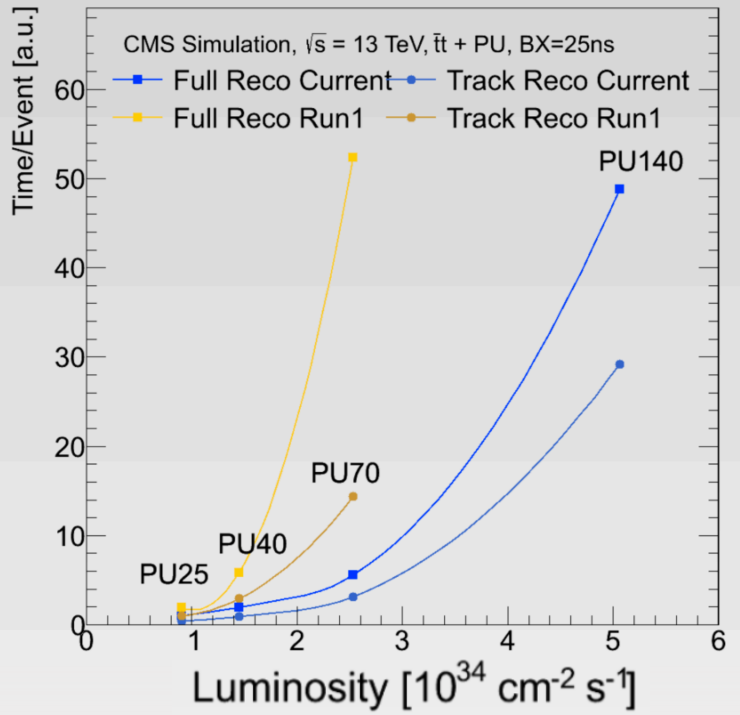


Length of the Prompt reco delay



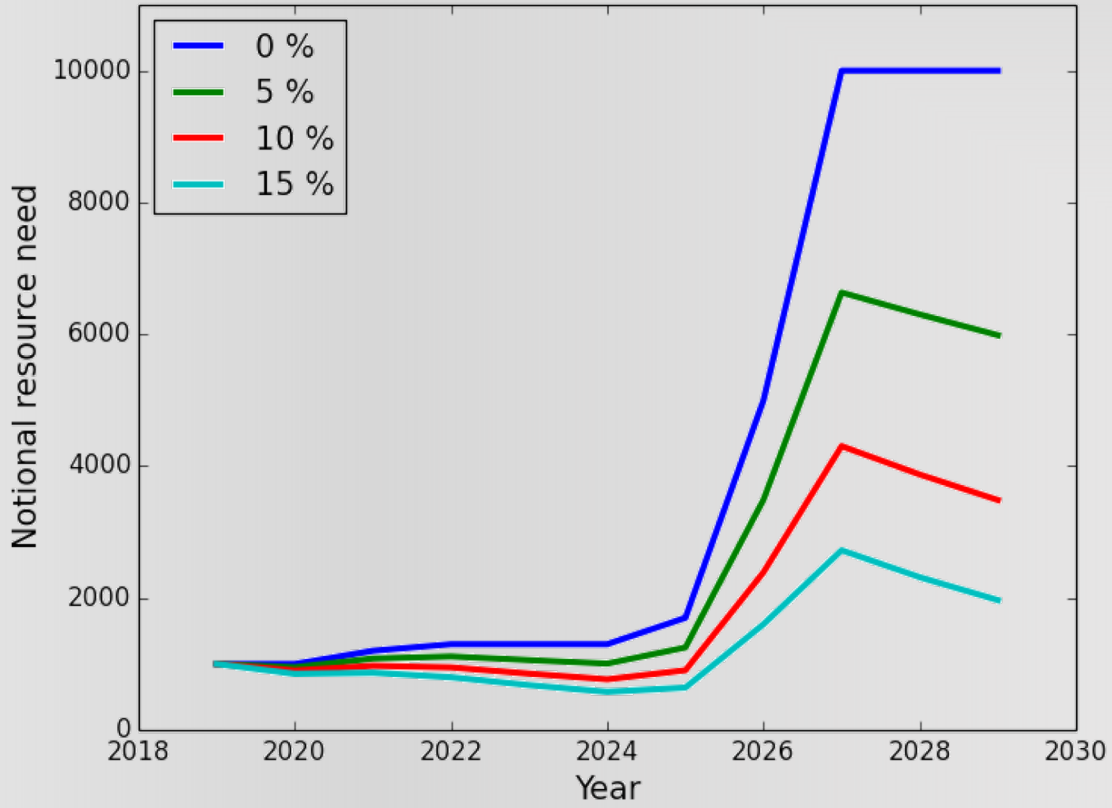
Clearly the uncertainties are large – what handles do we have to deal with these?

- Many factors enter resource modeling as exponentials. Examples
  - Assumptions about purchasing power improvements
  - Assumptions of code performance improvements
  - Estimates of time per event for some applications with respect to pileup (eg, reconstruction)
- Most application software in place today is quite preliminary or needs to evolve considerably to meet
  - This goes for everything from generators to detector reconstruction
  - In terms of both physics performance and technical performance
- Difficult to anticipate impact of R&D or behavioral changes before they happen
  - Models based on current practice plus “anticipated” improvements
  - Anticipate R&D outcomes by including “goals” for how R&D may change input parameters (examples: future data tier size reductions, application software speedup)

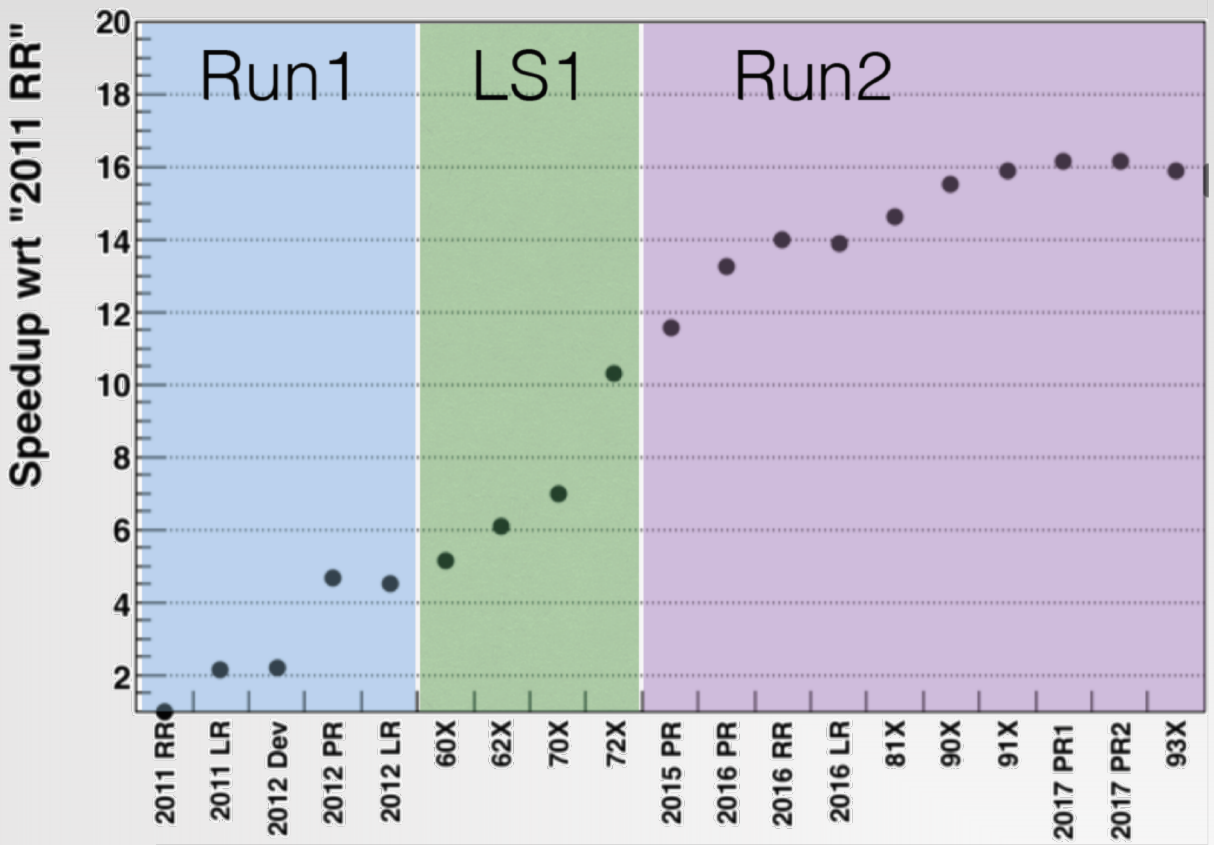


Variation in CPU resource needs with ”annual” code improvement factors

The difference in resource needs between a 5% and a 10% annual improvement is 30% in 2027

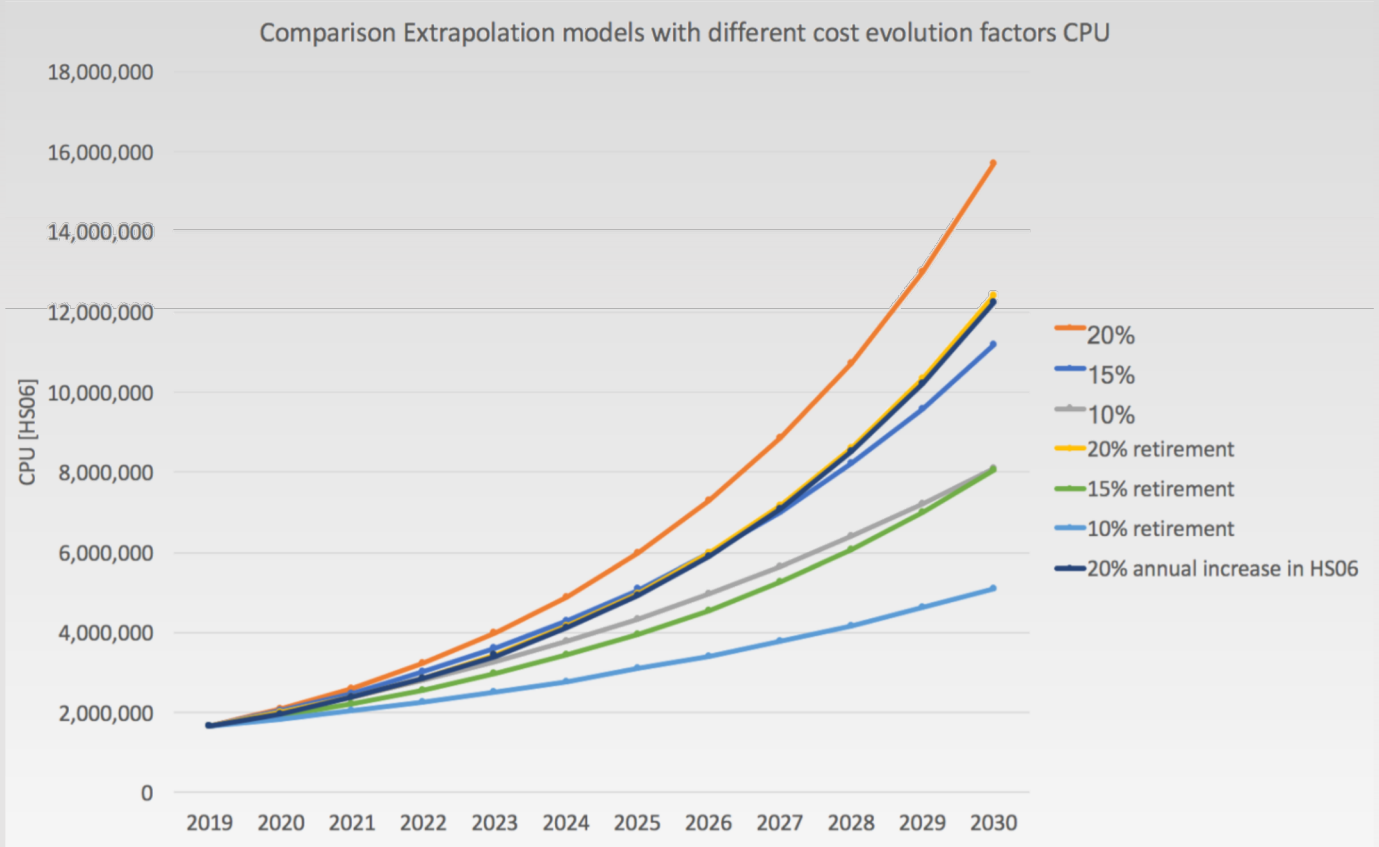


Annual speed ups have slowed in recent years. Algorithm reengineering efforts target HL-LHC timescales

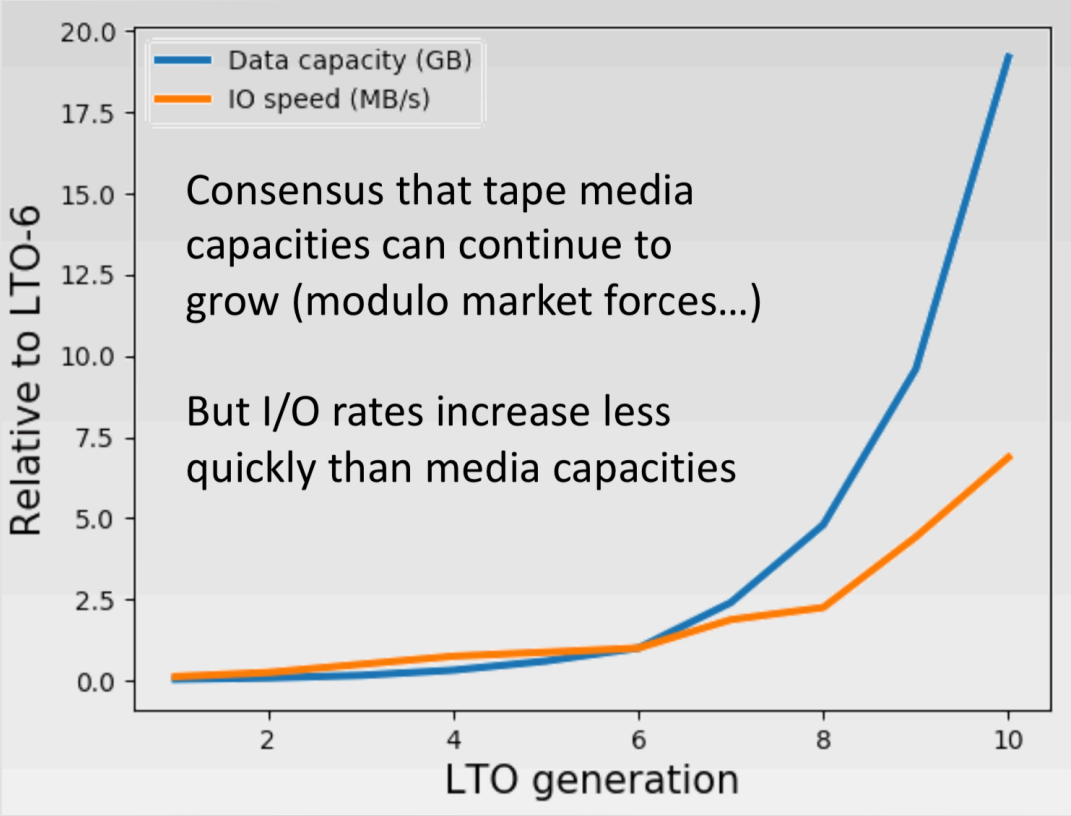


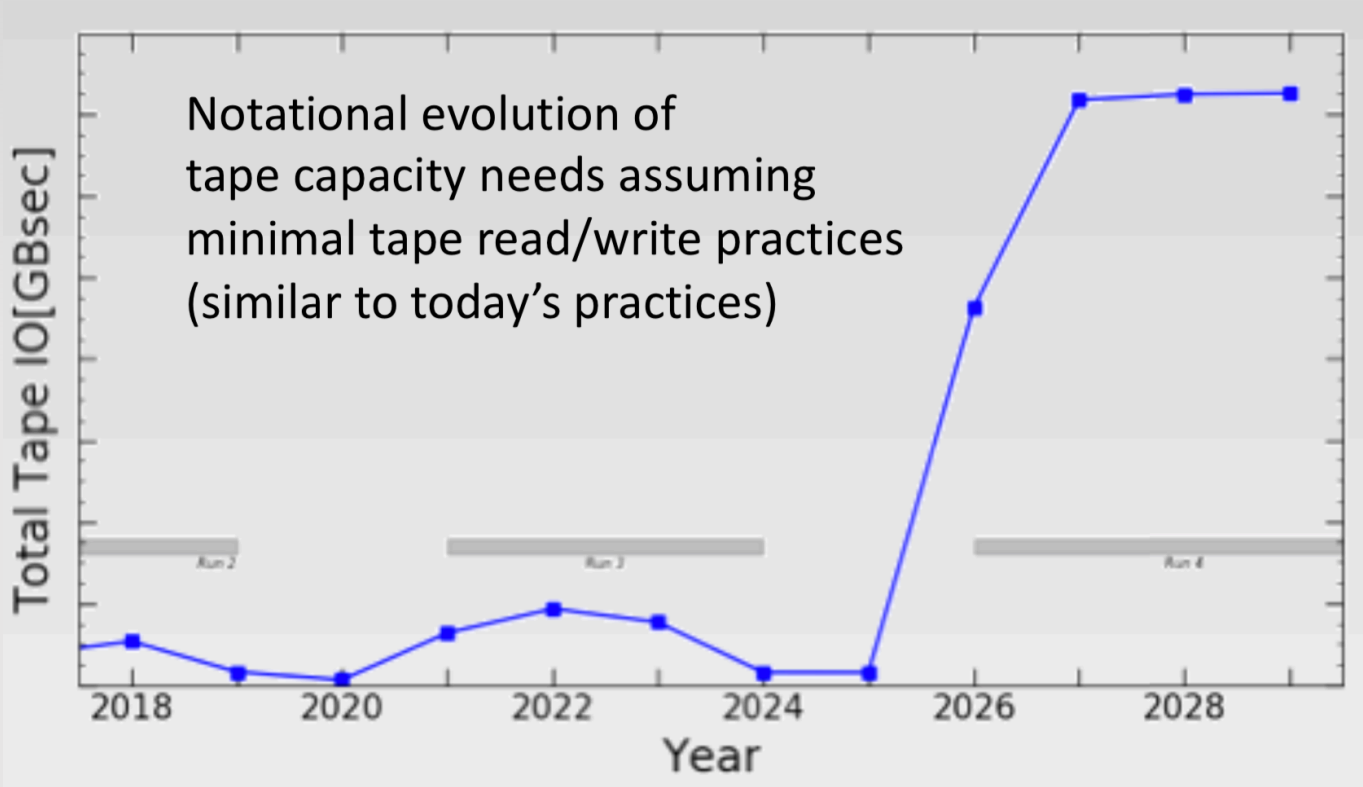
Uncertainties in cost evolution model (even without including fundamental changes in technology...)

- Cost reference planning is based on an assumed evolution of annual purchasing power.
- Implications are large as the effect enters exponetially
- Hardware retirements have substantial implications to “flat budget” unless the year over year growth is big



Including I/O from cold storage (aka, tape) is increasing important





- Tape I/O needs essentially scale the same way as tape storage needs

Back of the envelope tape recall calculations suggest 10x more drives to keep doing what we do today

- Calculation assumes CMS sites are using LTO-11 like hardward HL-LHC and can achieve a 80% read efficiency

	Run 2 GB/s	HL-LHC GB/s	# of LTO8/ Run2	# of LTO11/ HL-LHC
Write 1 copy of RAW during data taking	~0.3	~17	~1-2	~15
Write 1 copy of RAW+ derived during data taking	~0.6	~27	~2-3	~20
Recall 1 year of RAW in a month	~2.5	~140	~10	~120
Recall 1 year of AOD in a month	~1.1	~40	~5	~30

- Motivation for R&D towards cold-storage configurations supporting (especially) analysis data recall needs

Conclusions and looking forward

- Modeling used for short term / long term resource projections is maturing. Flexibility is key for adjusting to evolving interests
- Keep in mind how uncertainties increase with the length of projection. Major unexpected changes in approach or technology
- Tape I/O and network needs are examples of needed estimates that we will now routinely track
- As applications are beginning to integrate hardware accelerators - they are a clear next step.
  - Accelerators integrated into a server could be simply estimated via an effective “HS06” value (but assessed using a benchmark representative of CMS application performance)
  - Dedicated services need to treated separately