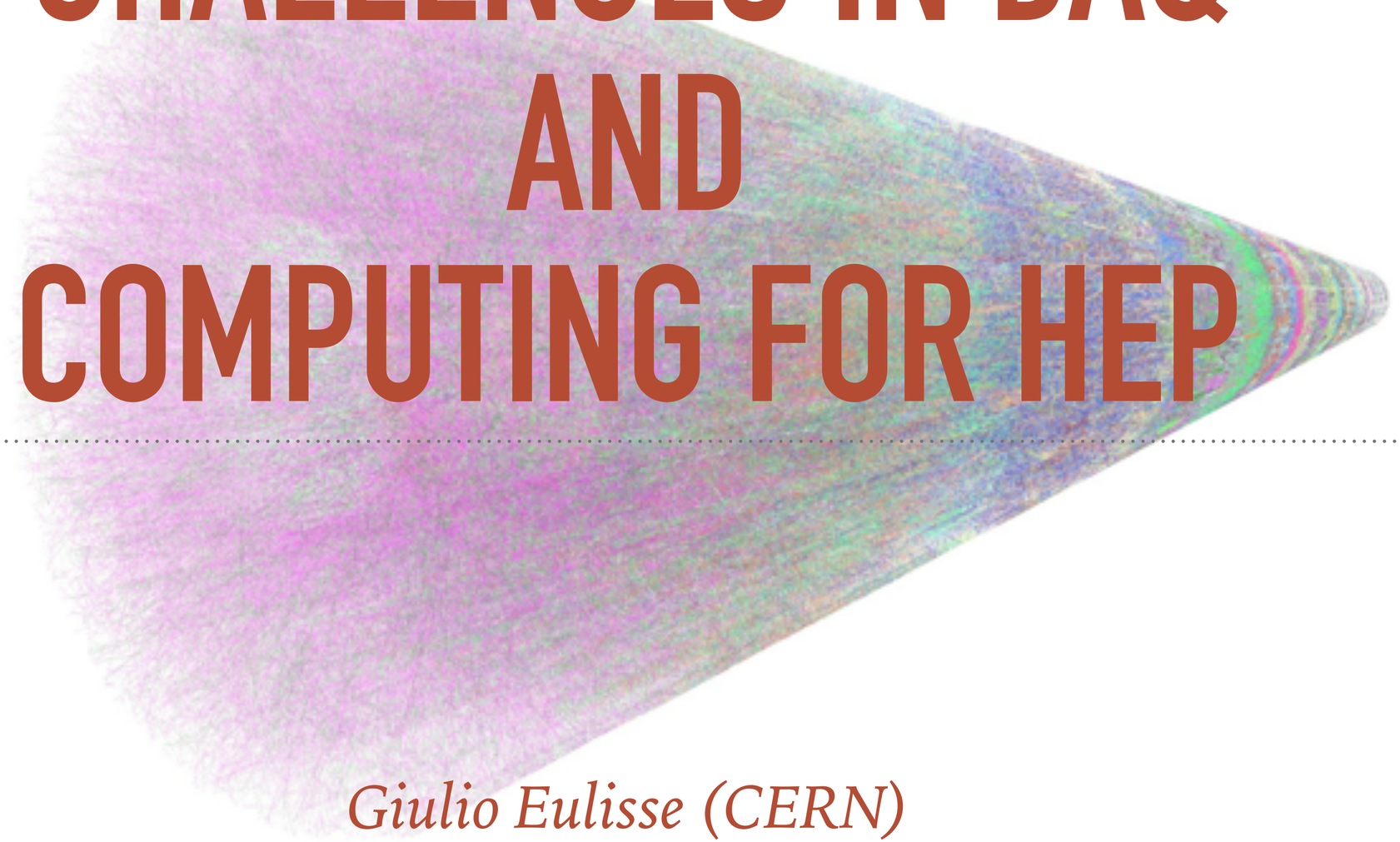
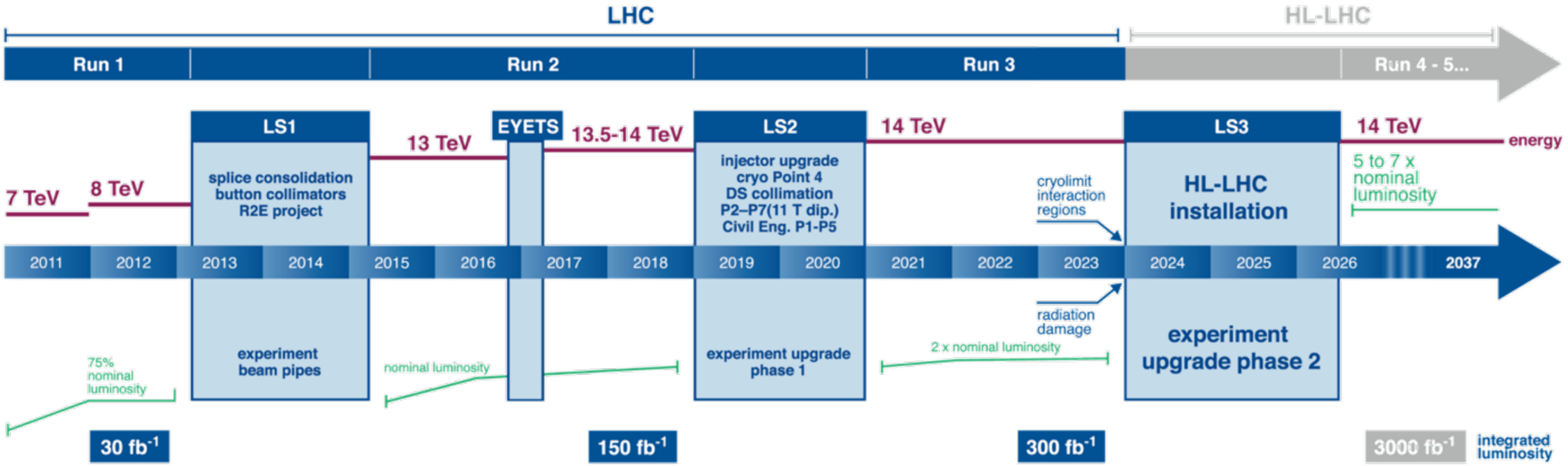
CHALLENGES IN DAQ AND COMPUTING FOR HEP



Giulio Eulisse (CERN)

LHC UPGRADE SCHEDULE



Courtesy David Rohr

***LS2 LHC upgrade**: heavy ion rate: >50 kHz in Run 3 (>10 kHz now), boost pp collision rate by small factor.*

***LS3 LHC upgrade**: HL-LHC era, boost pp collision rate by factor 5 – 7 in Run 4.*

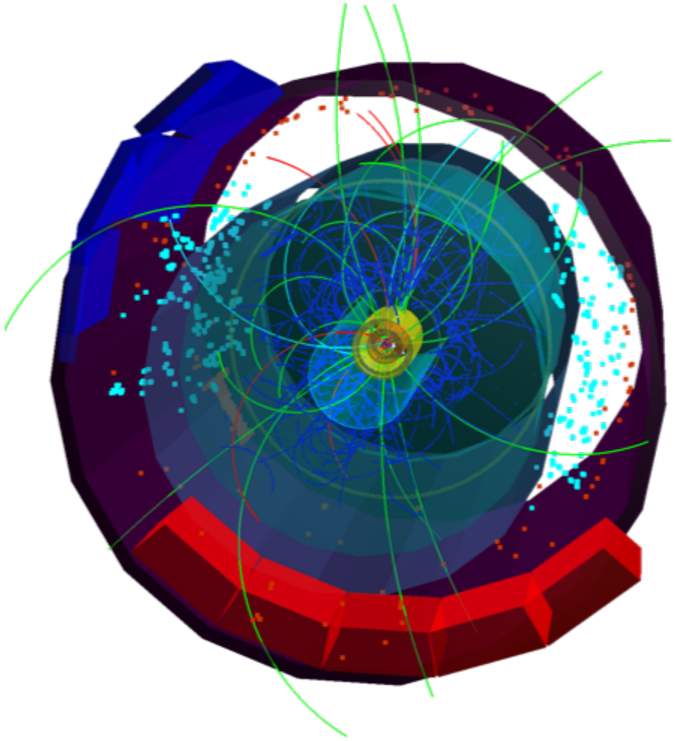
Highest pp luminosity only for ATLAS and CMS – their detectors are upgraded for Run 4 accordingly.

ALICE and LHCb perform a major upgrade for Run 3, basically now.

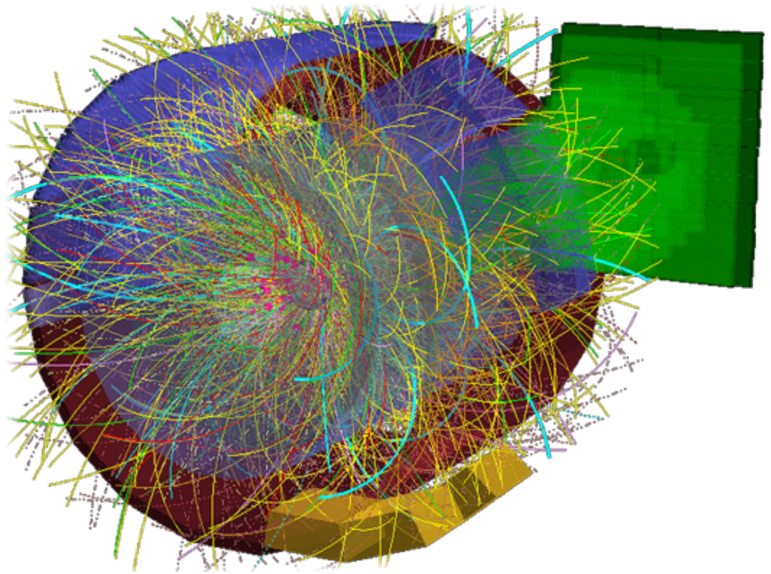
Run 3 is adiabatic increase for ATLAS / CMS, Run 4 will be adiabatic for LHCb with no increase for ALICE.

AN EXAMPLE: ALICE IN RUN 2

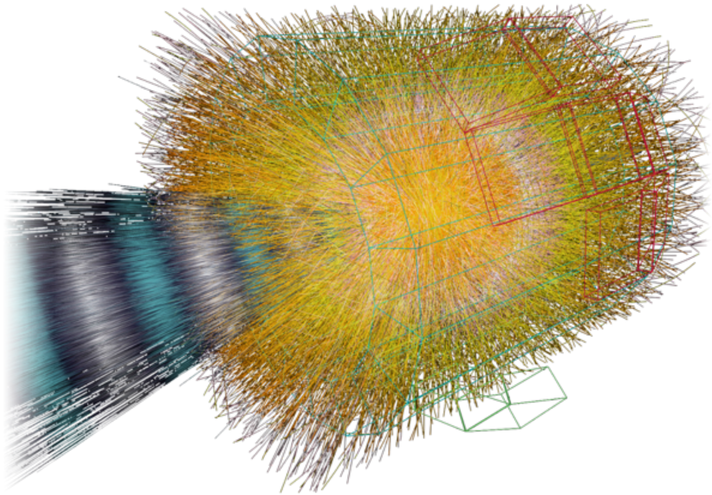
From O(1) kHz single events...



p - p



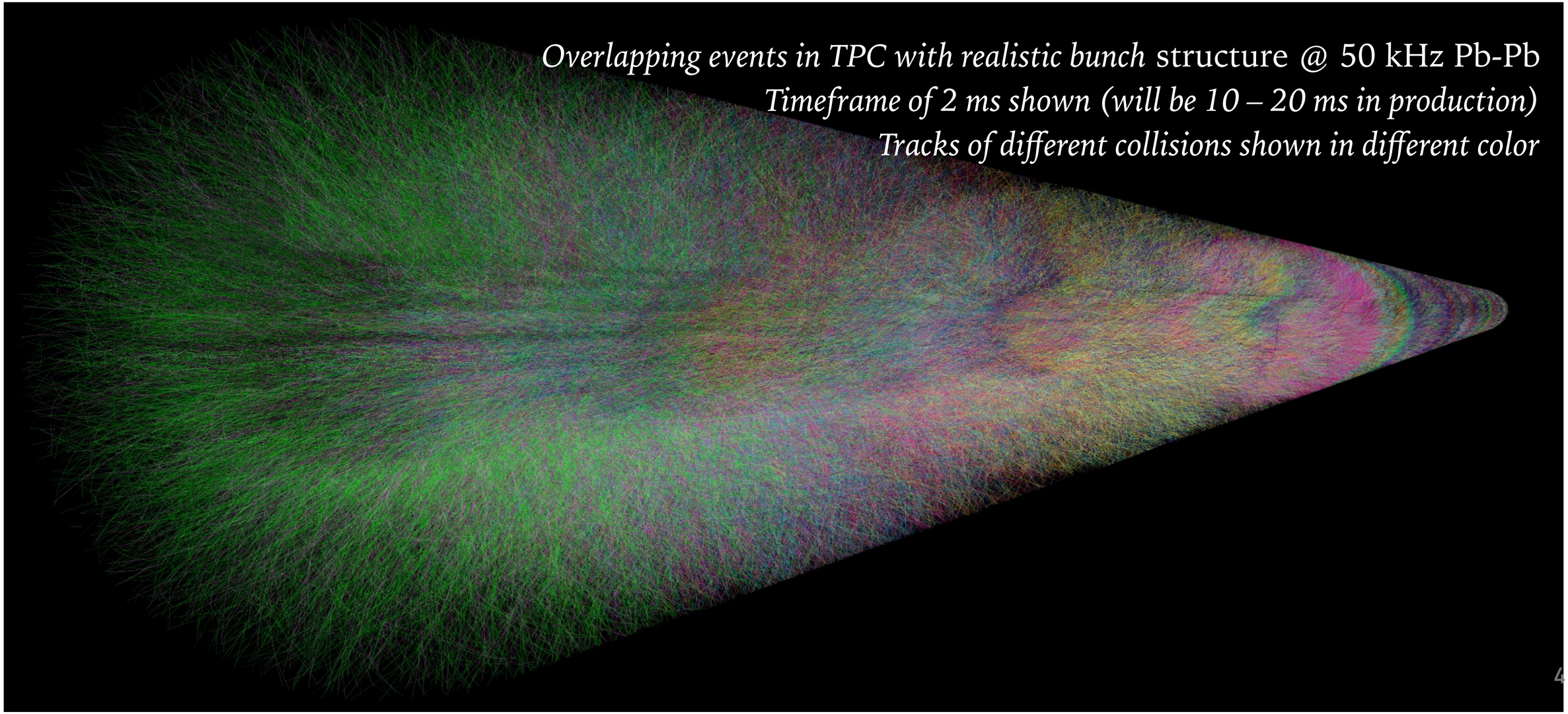
p - Pb



Pb - Pb

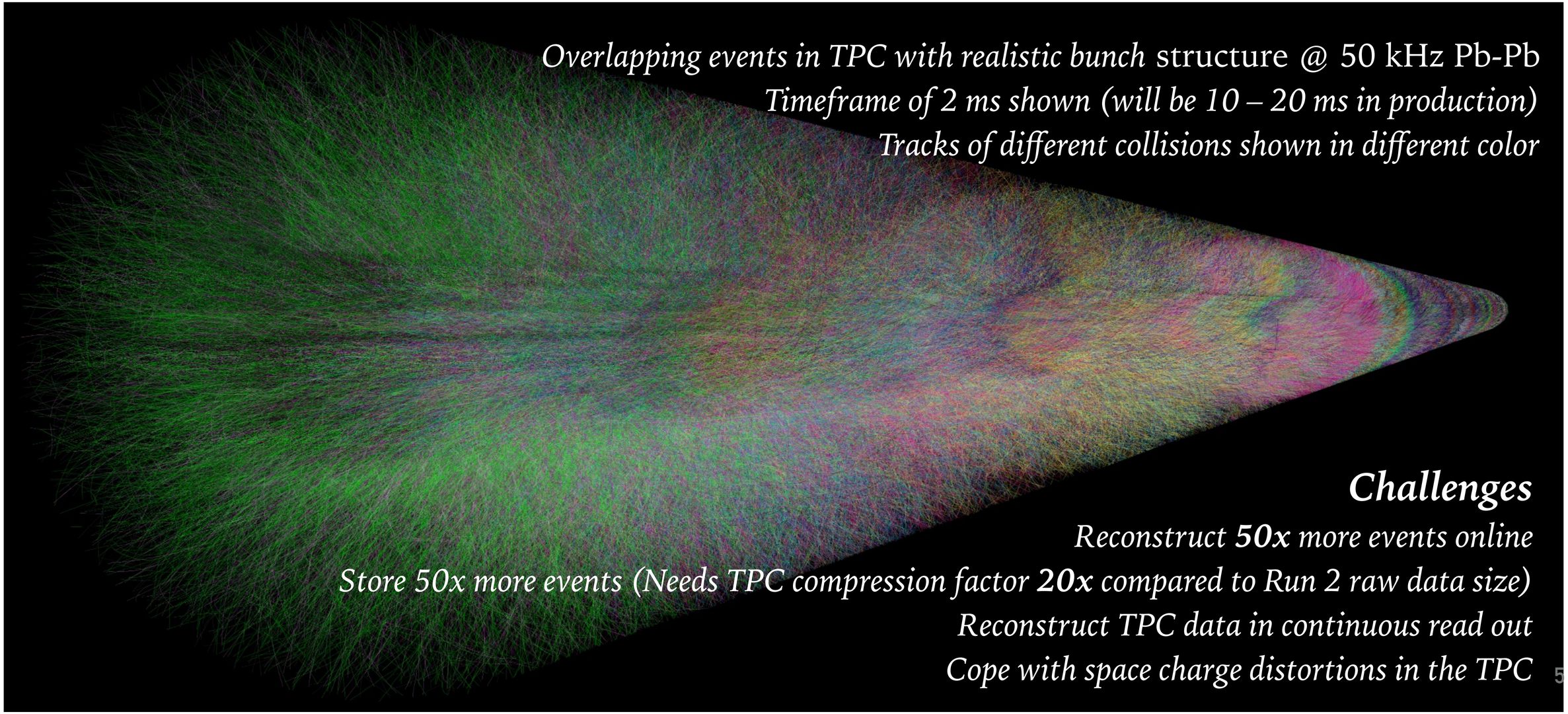
AN EXAMPLE: ALICE IN RUN 3

…to 50kHz of continuous readout data.

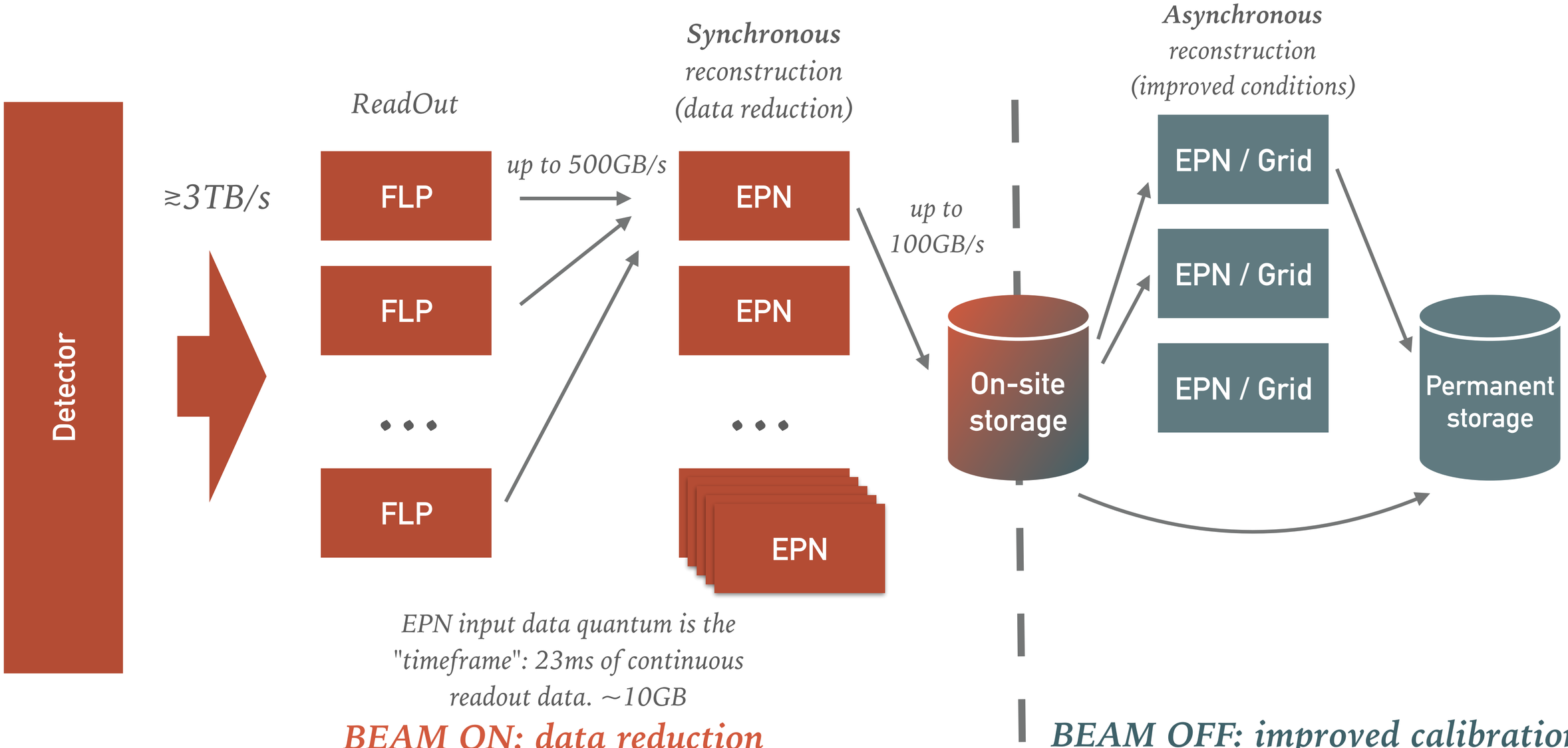


AN EXAMPLE: ALICE IN RUN 3

…to 50kHz of continuous readout data.



AN EXAMPLE: ALICE IN RUN 3



EPN input data quantum is the "timeframe": 23ms of continuous readout data. ~10GB

BEAM ON: data reduction

BEAM OFF: improved calibration

LHC DATATAKING OVERVIEW

	ALICE (Pb-Pb)		LHCb		ATLAS		CMS	
	Run 2	Run 3	Run 2	Run 3	Run 2 / 3	Run 4	Run 2 / 3	Run 4 (PU 140/200)
Luminosity	~10 kHz	50 kHz	$4*10^{32}$	$2*10^{33}$	$2.14*10^{34}$	$5–7.5 *10^{34}$	$2.14*10^{34}$	$5–7.5 *10^{34}$
Hardware trigger	500 Hz – 2 kHz	50 kHz continuous	1 MHz	- / Full 30 MHz bunch crossing rate	95 kHz	1 MHz (can evolve to 4)	100 kHz	500 / 750 kHz
HLT Accept	No rejection	No HLT	12.5 kHz	>100 kHz	1 kHz (< 2)	10 kHz	1 kHz	5 / 7.5 kHz
Raw Data Rate into HLT	45 GB/s (w. ZS)	3 TB/s (w.o. ZS)	55 GB/s	4 TB/s (w. ZS)	29 GB/s (260 GB/s L1)	2.6 TB/s (5.2 TB/s L1)	1.6 TB/s (event network)	23 / 44 TB/s (event network)
Data stored	~10 GB/s	Up to 100 GB/s	0.6 GB/s	2-10 GB/s	2.4 GB/s	50 GB/s	5 GB/s	32 / 61 GB/s
Data Buffer	~1 PB DAQ buffer to Tier0	~60 PB (one year of compressed data), up to 100 GB/s	~12 PT	~100 PB (two weeks of HLT1 accepted raw data, 150 + 150 GB/s read/write.	1.5 TB events + 48 hours to Tier 0	36 PB, 48 hours + L1 to HLT	12 TB (RAM disk, events before HLT, 60s)	171 / 333 TB (events before HLT, 60s)

Courtesy David Rohr

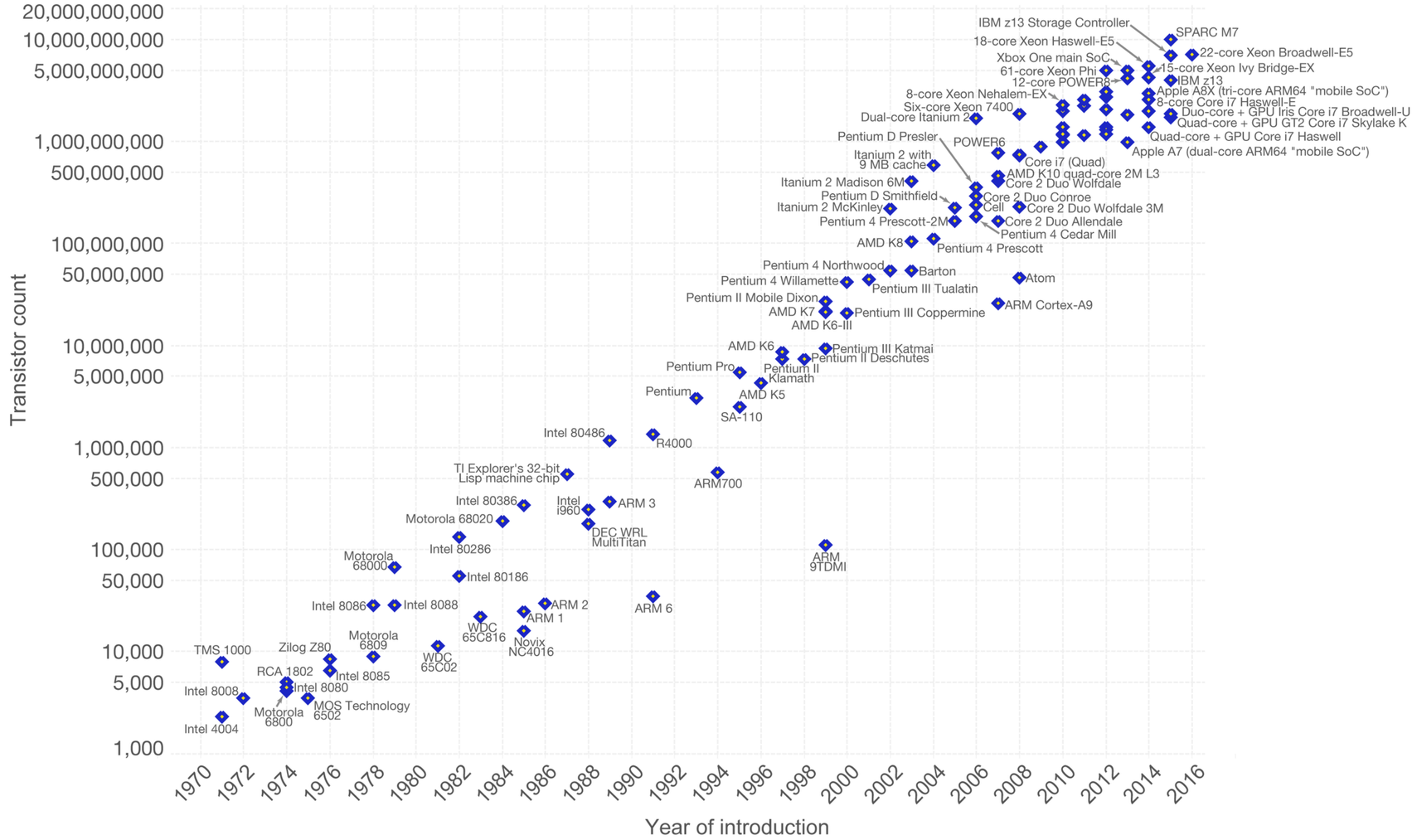
***ALICE** will take 50x more events, but only minimum bias, all other experiments collect 10x more statistics.*

***ALICE and LHCb** process all data in software, use large disk buffers to hold large amount of (compressed raw data) for processing in the online farm when there is no beam.*

***ATLAS and CMS** have much higher luminosity, full readout of front-end at bunch-crossing rate and processing in software not feasible and not cost effective.*

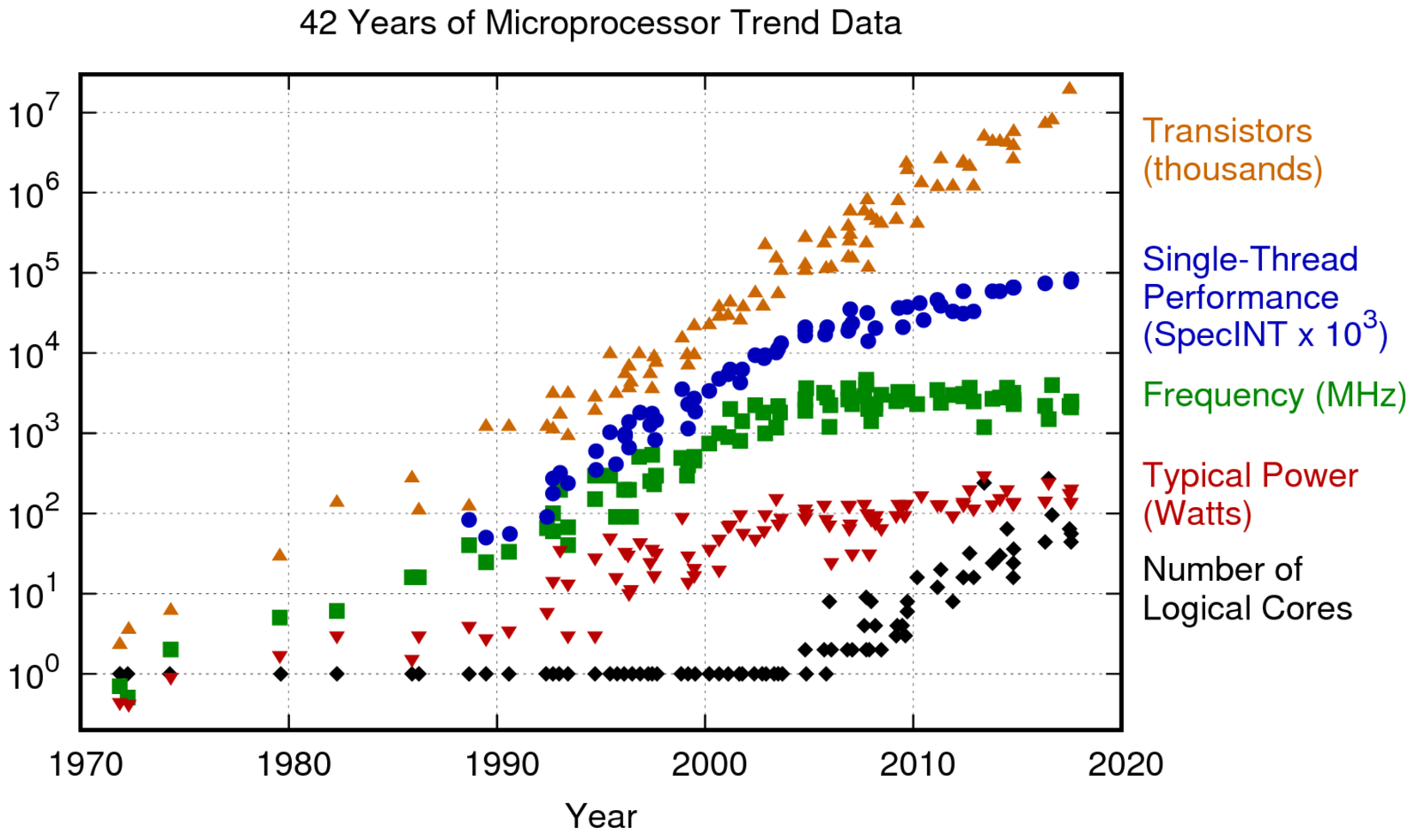
***ALICE** features high data rate during Pb-Pb (due to TPC), collects large amount of data in only few weeks.*

HOW TO COPE WITH THE CHALLENGE: MOORE LAW STILL HOLDS STRONG...



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

… TOO BAD IT USES THE WRONG OBSERVABLE.



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Moore's law is expressed in terms of ***transistor count per square inch****, not in terms of CPU frequency, nor single core performance.*

Limiting factor is the ability to dissipate heat.

If we want to take advantage of improvements in computing hardware, ***we must parallelise our payloads.***

EXPLOITING PARALLELISM: MULTICORE CPUS

Mainstream

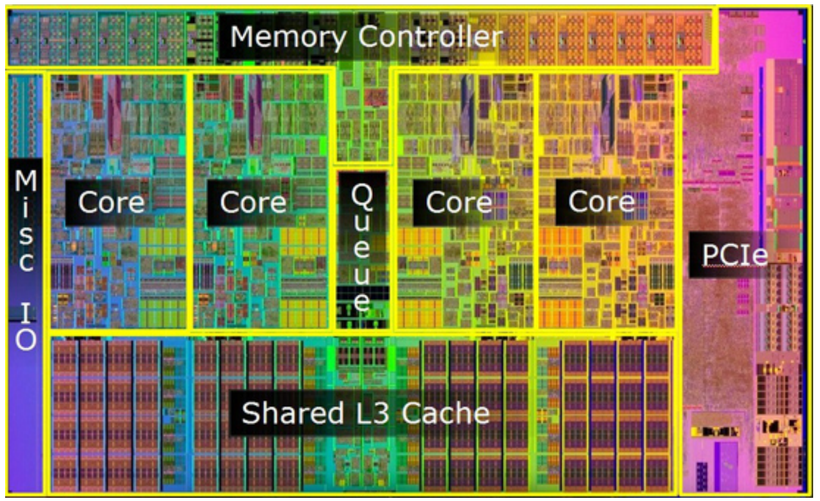
Product offerings from multiple vendors since over 10 years. Coupled with strong push on vectorisation it allowed industry to still scale global CPU performance over the years.

Very flexible

Single cores still powerful enough to do the full job. Low flop-per-byte requirements and easy to adopt / adapt programming model (provided you have enough memory).

Still learning phase

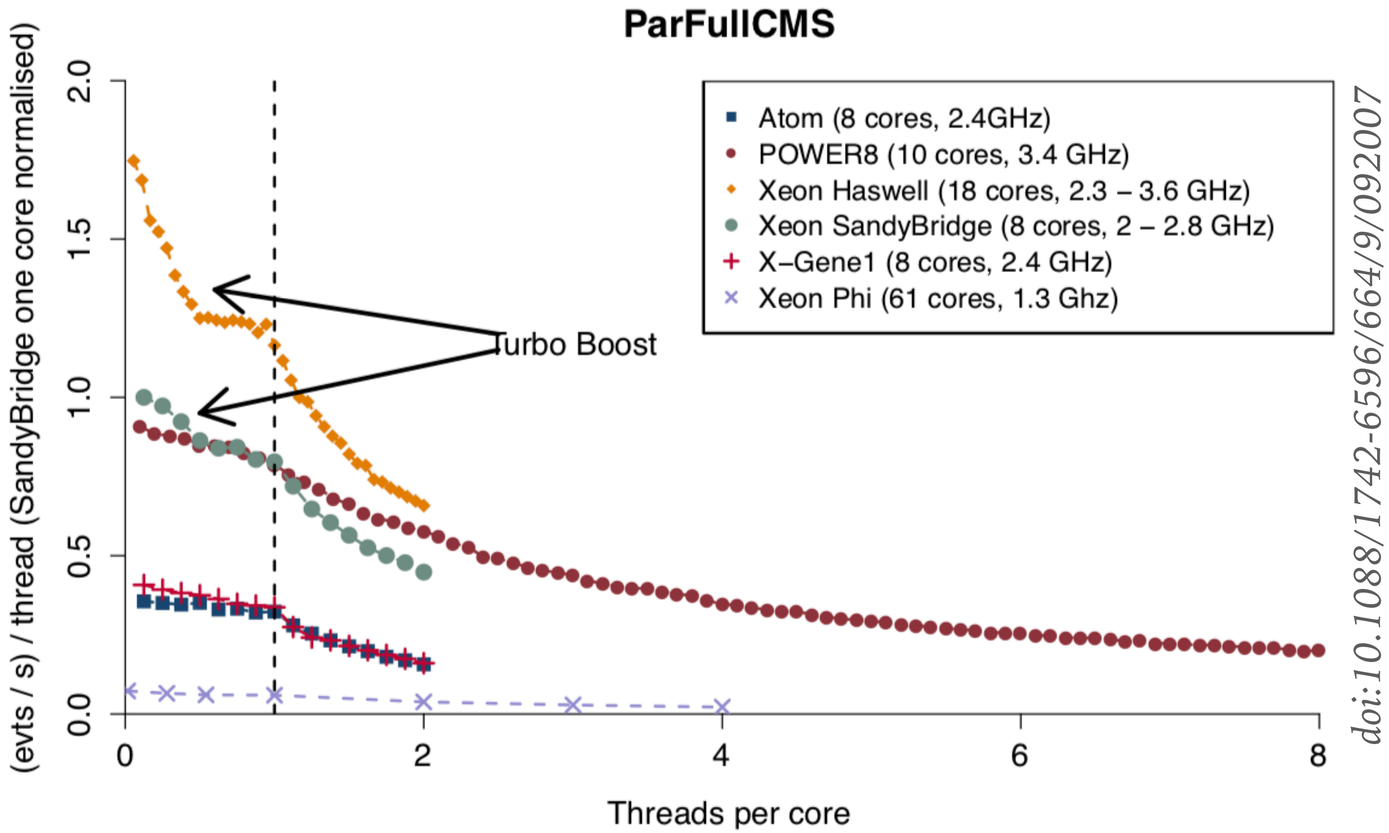
*We have done great steps forward but exploiting multiple CPUs is **not an easy task**. Main problem for HEP has been **limiting memory usage per core** (and **not dead-lock** while doing so). Serial bottlenecks (**initialisation, locked sections**) and resource contention (**caches, memory bus, I/O**) main issue when optimising for speed.*



$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

Amdahl's law: *as you speedup thanks to additional resources (s) your global speedup is limited by the serial fraction (p)*

MULTICORE CPUS: NOT IMMUNE TO TRADEOFFS



Aggressive power management *(e.g. Intel TurboBoost) makes peak performance only available for when running a reduced number of cores. CPUs might decide to* ***throttle frequency when using power hungry vector units (AVX2).***

FRAMEWORKS FOR A MULTICORE ERA

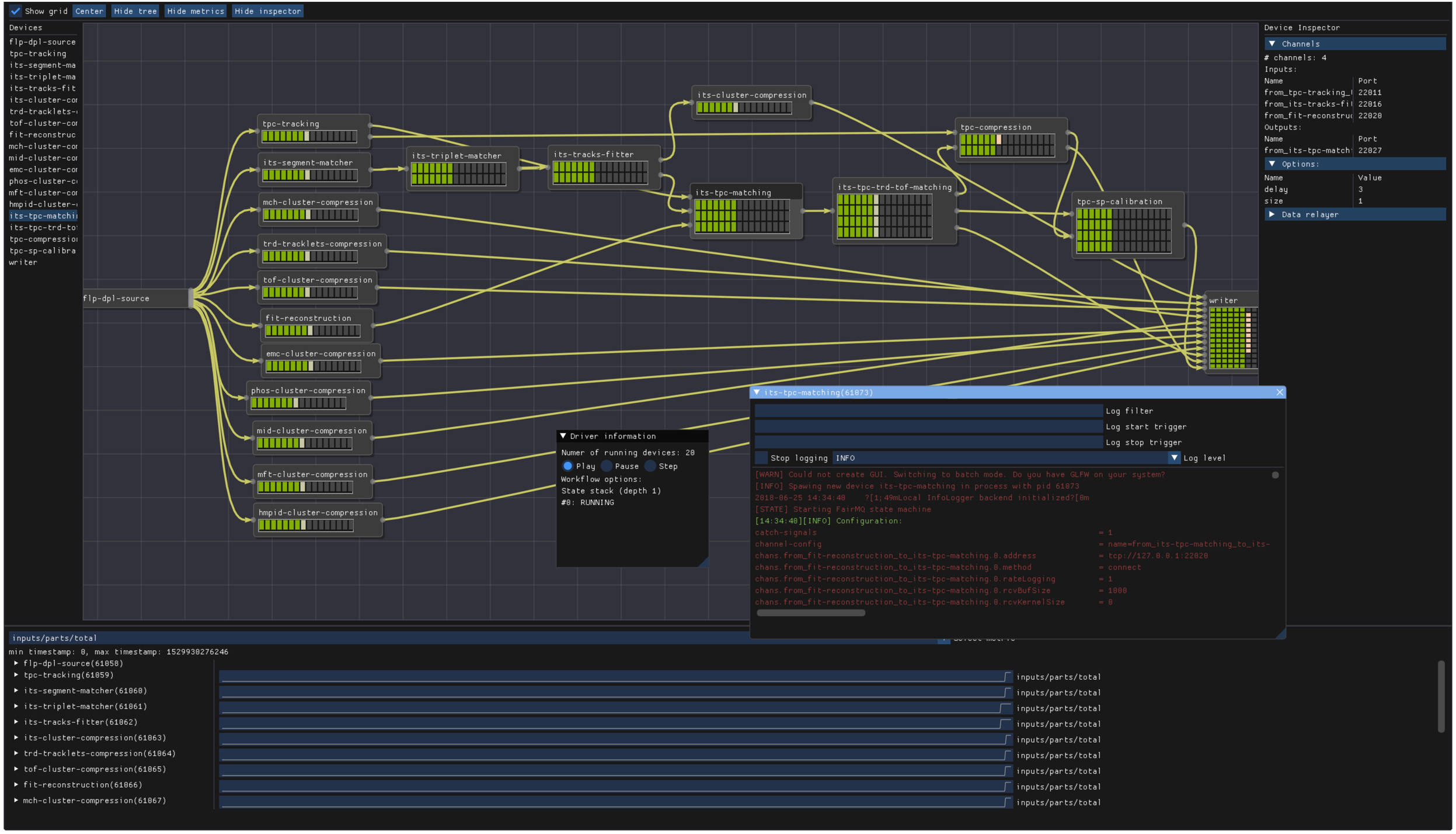
All LHC experiments have been adapting / evolving / revolutionising their software frameworks to be able to process data in a multicore world. There are two well established approaches:

Communicate by sharing:

Task based frameworks. E.g. CMSSW, Gaudi Hive. Parallelism is expressed as a set of tasks executed in parallel by worker threads, transforming a shared object hierarchy. Naturally represents a (micro)batch system.

Share by communicating:

Data streaming frameworks. E.g. FairMQ / ALICE O2, JLAB CLARA. Parallelism is expressed as Communicating Sequential Processes, exchanging messages. Lock free. Naturally fits heterogeneous environments, by modelling them as separate sets of processes.


Show grid
Center
Hide tree
Hide metrics
Hide inspector
Devices
flp-dpl-source
tpc-tracking
its-segment-ma
its-triplet-ma
its-tracks-fit
its-cluster-cor
trd-tracklets-
tof-cluster-cor
fit-reconstruc
mch-cluster-cor
mid-cluster-cor
emc-cluster-cor
phos-cluster-c
mft-cluster-cor
hmpid-cluster-
its-tpc-matchi
its-tpc-trd-to
tpc-compressio
tpc-sp-calibra
writer
flp-dpl-source
tpc-tracking
its-segment-matcher
its-triplet-matcher
its-tracks-fitter
its-cluster-compression
its-tpc-matching
its-tpc-trd-tof-matching
tpc-compression
tpc-sp-calibration
writer
mch-cluster-compression
trd-tracklets-compression
tof-cluster-compression
fit-reconstruction
emc-cluster-compression
phos-cluster-compression
mid-cluster-compression
mft-cluster-compression
hmpid-cluster-compression
Driver information
Numer of running devices: 20
Play
Pause
Step
Workflow options:
State stack (depth 1)
#0: RUNNING
its-tpc-matching(61073)
Log filter
Log start trigger
Log stop trigger
Stop logging
INFO
Log level
[WARN] Could not create GUI. Switching to batch mode. Do you have GLFW on your system?
[INFO] Spawing new device its-tpc-matching in process with pid 61073
2018-06-25 14:34:40 ?[1;49mLocal InfoLogger backend initialized?[0m
[STATE] Starting FairMQ state machine
[14:34:40][INFO] Configuration:
catch-signals = 1
channel-config = name=from_its-tpc-matching_to_its-
chans.from_fit-reconstruction_to_its-tpc-matching.0.address = tcp://127.0.0.1:22020
chans.from_fit-reconstruction_to_its-tpc-matching.0.method = connect
chans.from_fit-reconstruction_to_its-tpc-matching.0.rateLogging = 1
chans.from_fit-reconstruction_to_its-tpc-matching.0.rcvBufSize = 1000
chans.from_fit-reconstruction_to_its-tpc-matching.0.rcvKernelSize = 0
Device Inspector
Channels
channels: 4
Inputs:
Name
Port
from_tpc-tracking_
22011
from_its-tracks-fi
22016
from_fit-reconstru
22020
Outputs:
Name
Port
from_its-tpc-match
22027
Options:
Name
Value
delay
3
size
1
Data relayer
inputs/parts/total
min timestamp: 0, max timestamp: 1529930276246
flp-dpl-source(61058)
tpc-tracking(61059)
its-segment-matcher(61060)
its-triplet-matcher(61061)
its-tracks-fitter(61062)
its-cluster-compression(61063)
trd-tracklets-compression(61064)
tof-cluster-compression(61065)
fit-reconstruction(61066)
mch-cluster-compression(61067)
inputs/parts/total

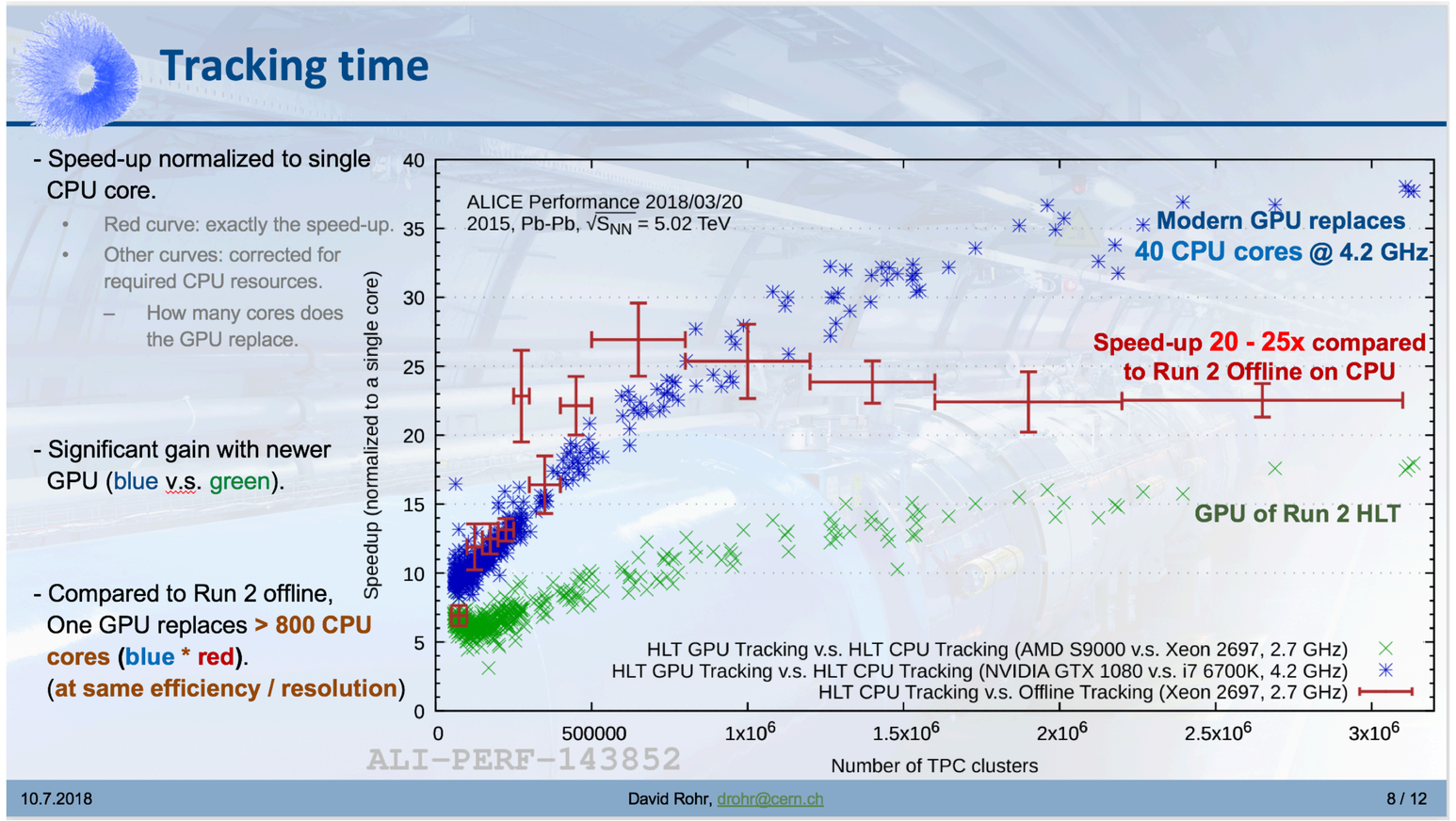
EXPLOITING PARALLELISM: GPUS

Videogames / movies industry (and lately Bitcoin mining) has gifted us with highly cost effective solutions for massively parallel problems, like drawing pixels...

Realtime raytracing demo by Epic Games, NVIDIA and ILMxLAB

EXPLOITING PARALLELISM: GPUS

... or doing TPC tracking in ALICE for Run 3. One modern GPU replaces 40 CPU cores. Changing the algorithm gives an additional 20x - 25x speedup with comparable quality.



David Rohr @CHEP 2018

EXPLOITING GPUS: EXAMPLE ALICE TPC TRACKING

ALICE TPC Track reconstruction for Run 3 derived from Run 2 Online Tracking.

Cellular Automaton + Kalman Filter. CMS is looking in similar approach.

Runs on GPUs (Common source code for CPU / GPU with OpenMP / CUDA / OpenCL).

See full talk by David @ CHEP for full details

Enormous speed-up compared to Run 2 offline.

***20x – 25x** speed-up on single CPU core.*

GTX 1080 GPU replaces ~800 CPU cores (running Run 2 offline code).

Processing of 23 ms time frame needs ~20 seconds on one EPN. (Compute farm has ~1500 EPNs).

Tracking independent from absolute z-position (needed to process time frames).

Same efficiency and resolution as Run 2 offline (some decline for deep secondaries).

Small decline in efficiency for short low-pT secondaries with 50 kHz timeframes as compared to single events(unavoidable due to higher occupancy).

Need TPC data compression factor 20x (compared to Run 2 raw data size).

Factor 8.3 in Run 2, Run 3 prototype achieves 9.1.

Potential to gain missing factor 2 by removing clusters not used for physics (removal of clusters of low-pT tracks down to 10 MeV/c already working in tracking).

GPUS: CHALLENGES

Different skill set

Requires a different / more advanced skill-set compared to a traditional "physicist programmer". Needs rethinking of CS training for physicists. Providing frameworks to simplify job by composing smaller building blocks is key for large scale adoption by community.

Closed environment

Developing environments on GPUs are traditionally more variegated (e.g. NVIDIA's CUDA, OpenCL / Vulkan, Apple's Metal, Microsoft DX) and vendor controlled compared to CPUs ones.

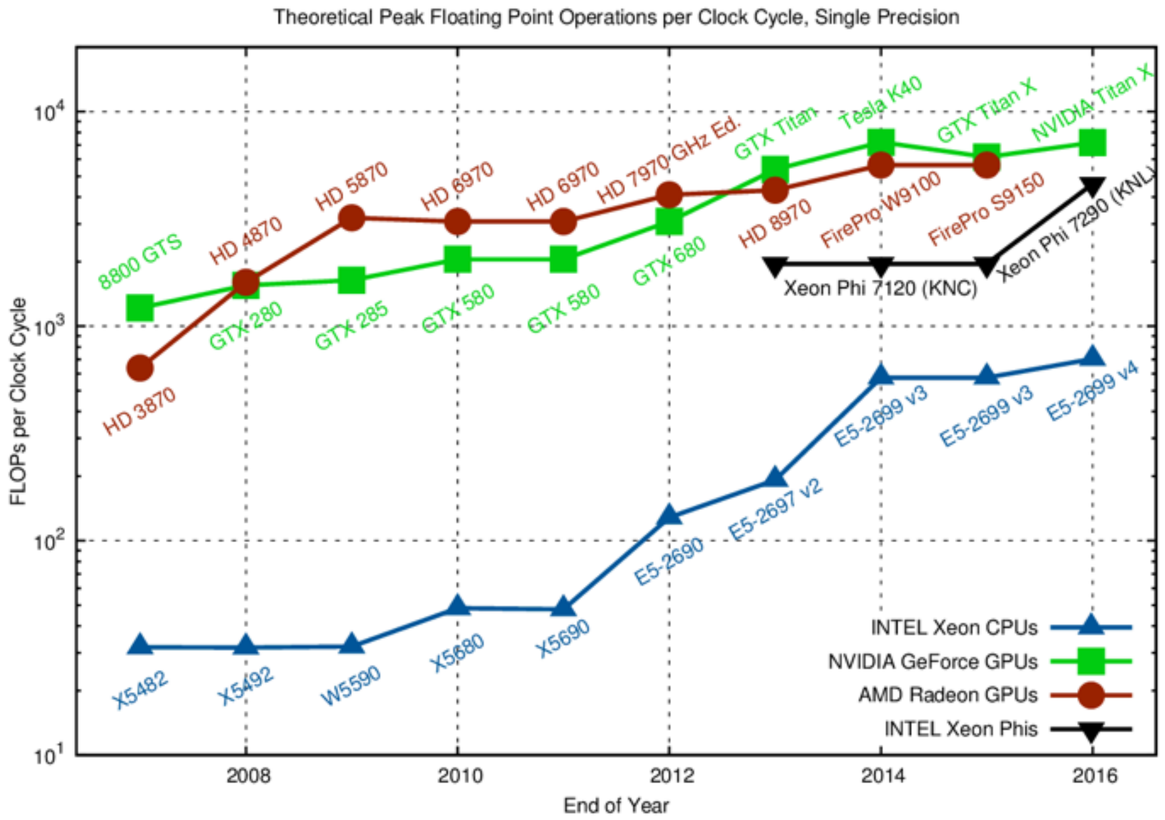
Deployment challenges

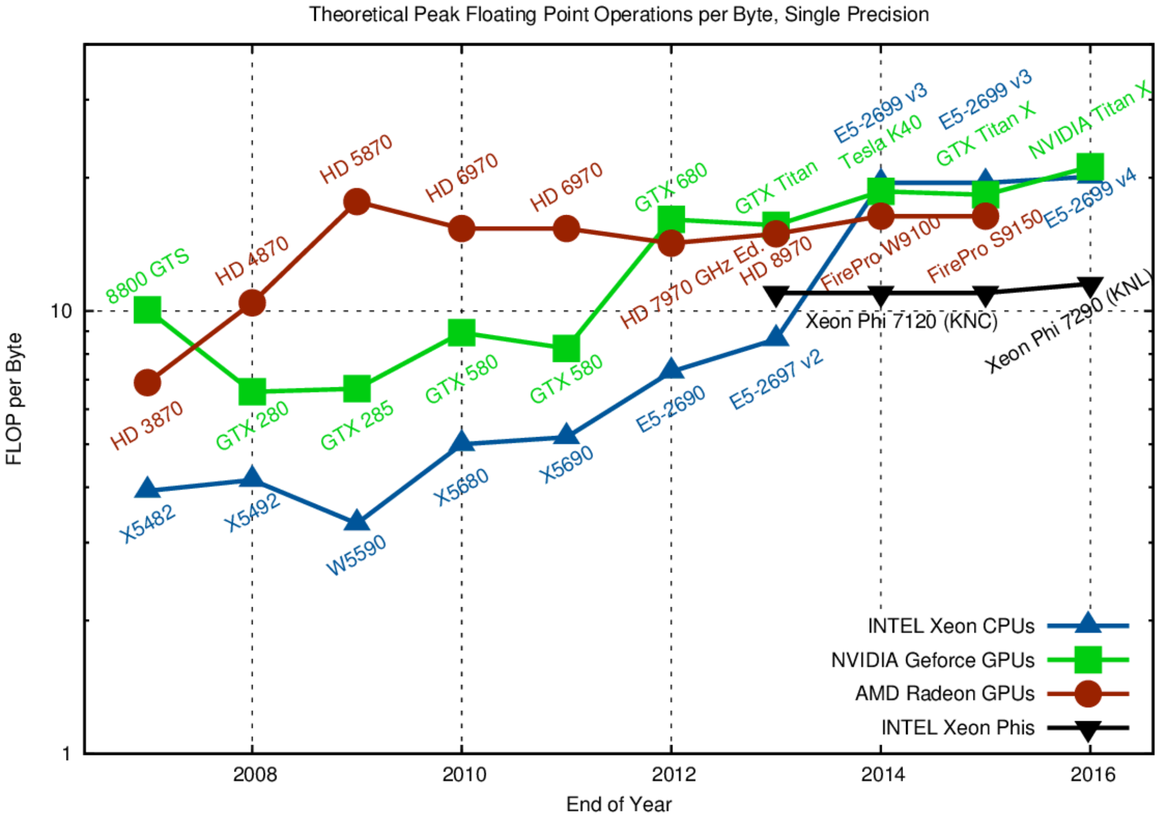
Hardware purchases, software licensing, retrofitting of existing infrastructure, deployment on the Grid, resource accounting.

Clear gains required in order to offset difficulties

GPUS: CHALLENGES

Different algorithms: ***memory bandwidth and latency are the limiting factor.*** *Algorithms with a lot of branching (tracking!) are less suitable. Reaching GPU full potential requires algorithm FLOPs/bytes read ratio is high (i.e. computation bound).*

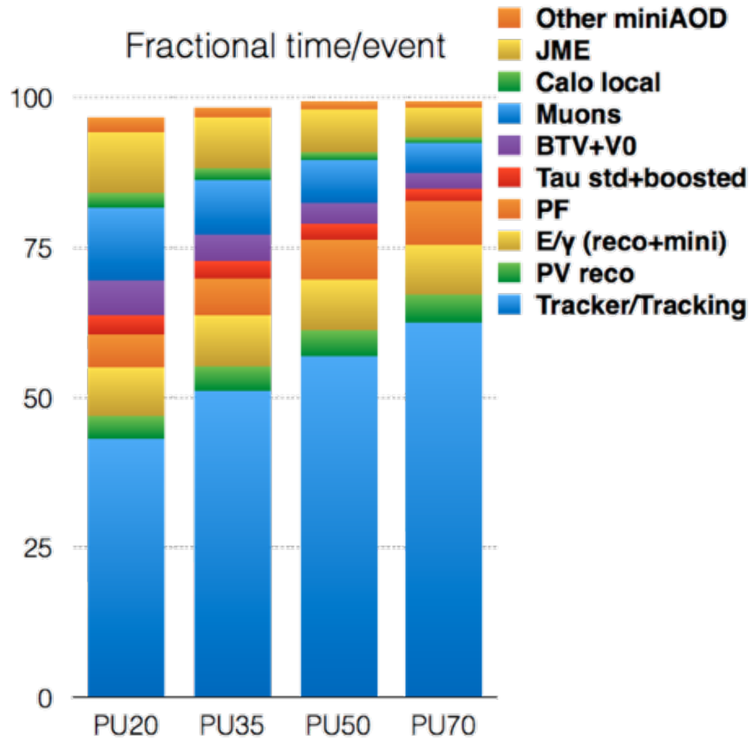


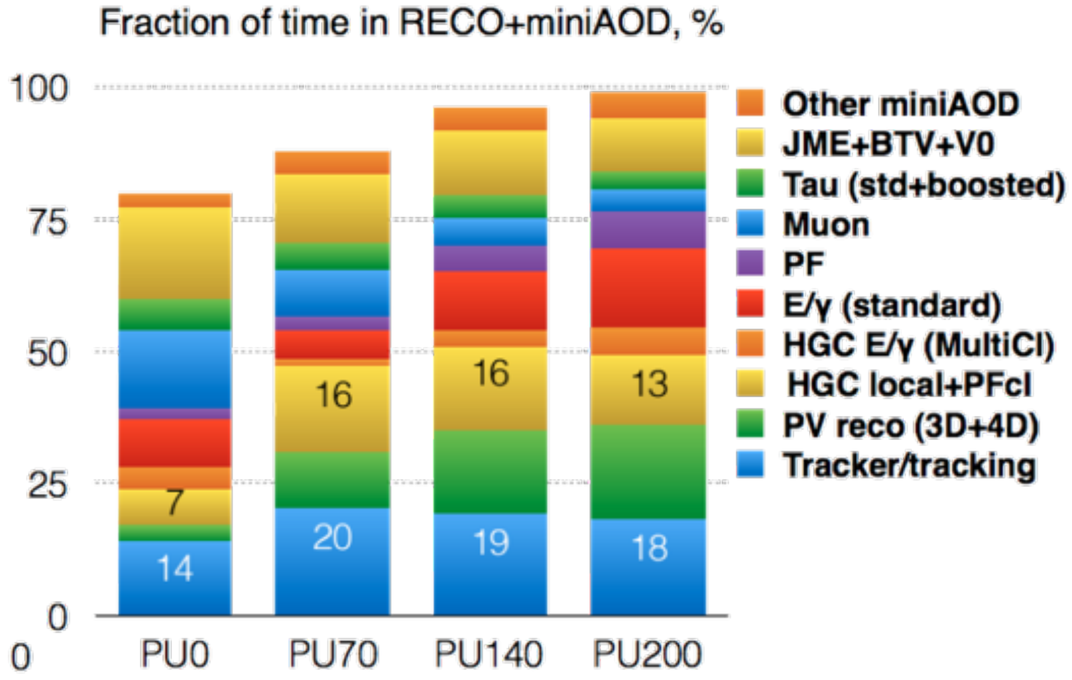


"Good" news is CPUs are increasingly memory starved as well. Optimisation efforts for GPU are useful to fully exploit CPUs as well.

MORE EXAMPLES: CMS IN RUN 4

Courtesy Tommaso Boccali





Run I + II + III:

Tracking is the most time consuming algorithm; it also shows the strongest scaling with LHC PileUp.

In principle, optimise tracking and you are ok!

Run IV:

New calorimeters have much larger number of channels.

Optimising tracking no more sufficient!

CMS IN RUN 4: HOW TO GET AROUND?

Courtesy Tommaso Boccali

AI (ML, DL, …) is increasingly being tested as the paradigm for faster and more accurate algorithms in reconstruction.

GPU at the trigger level (HLT) are in the plans. Less clear offline, needs Framework R&D in order to better treat heterogeneity.

Work with Geant to study GeantV – transition to be decided by Run III.

CMS is currently the most advanced in the deployment of small data formats for analysis.

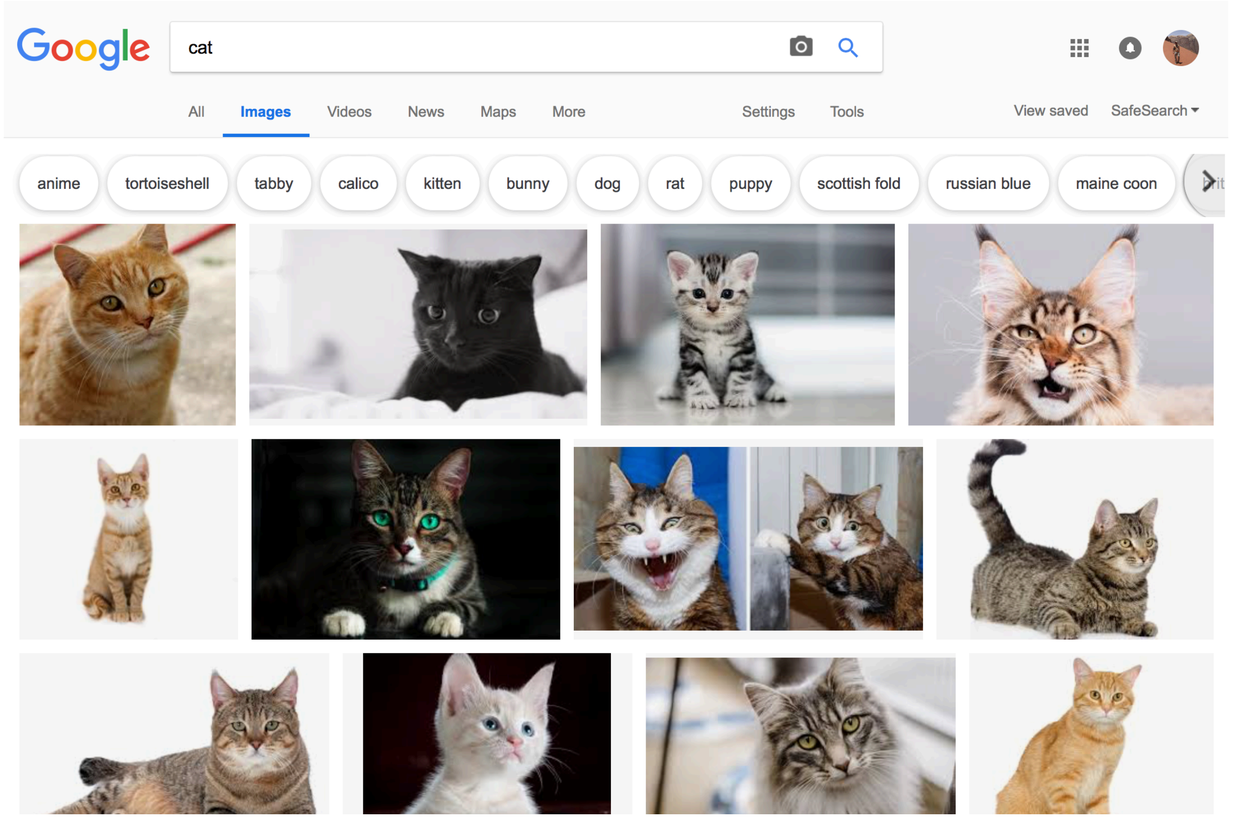
AOD (RunI analysis format) = 400 kB/ev;
MiniAOD (RunII) = 50 kB/ev;
NanoAOD (RunII - RunIII) = 1 kB/ev.

Data-lakes as a way to increase storage efficiency and to increase the utilisation of Opportunistic/Storage less/HPC resources.

MACHINE LEARNING (DEEP LEARNING)

Hype of the moment *(and of ~40 years ago, but we do learn from mistakes, don't we?)*

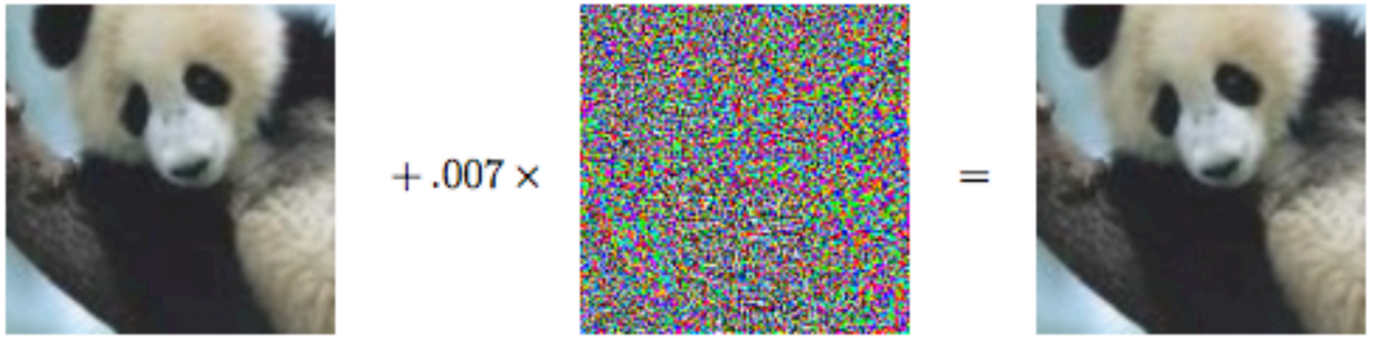
Finds cats pictures for you…



MACHINE LEARNING (DEEP LEARNING)

Hype of the moment *(and of ~40 years ago, but we do learn from mistakes, don't we?)*

Finds cats pictures for you... when not fooled...

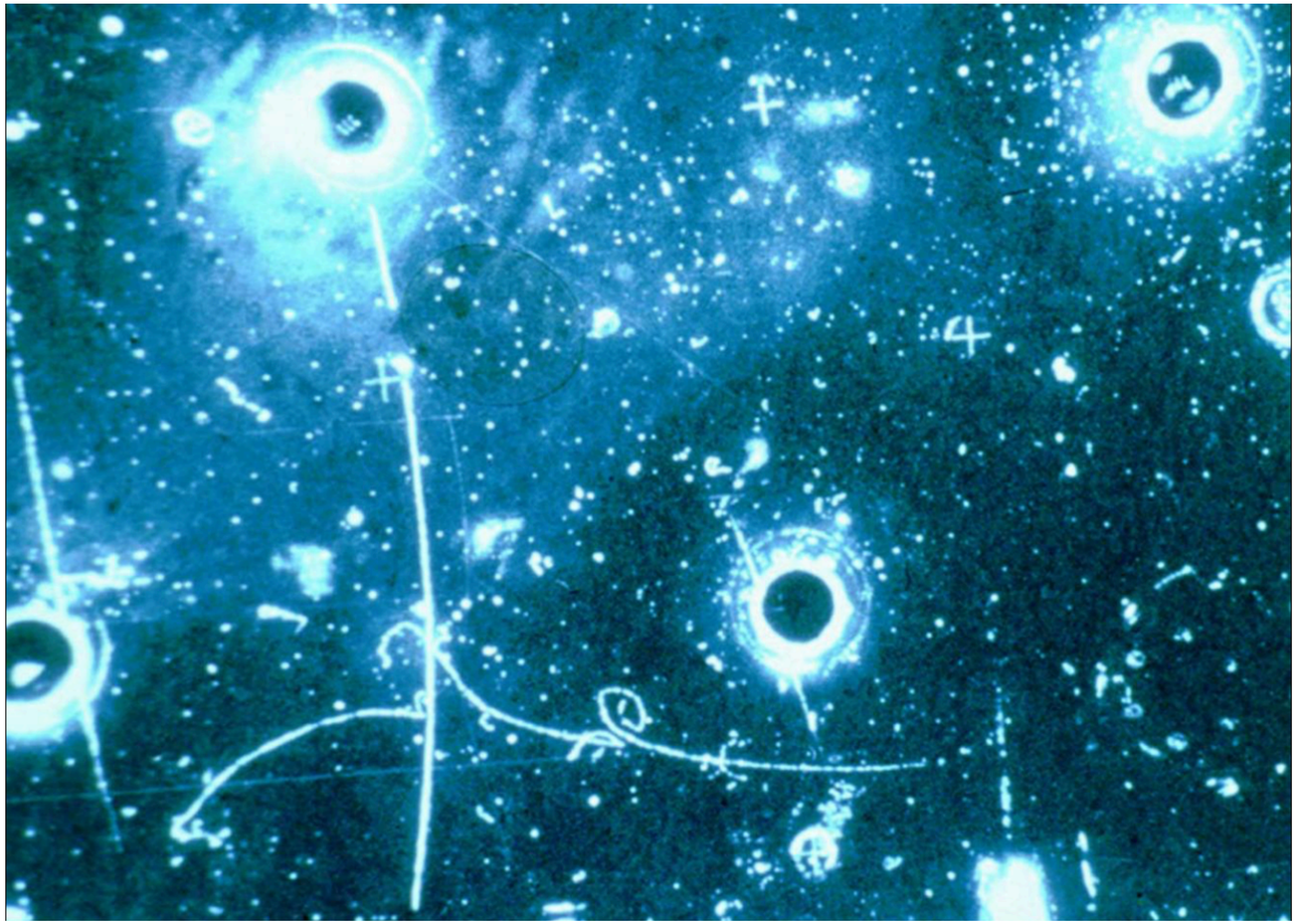


https://arxiv.org/abs/1412.6572

MACHINE LEARNING (DEEP LEARNING)

Major R&D field also in HEP.

Tracking & pattern recognition: *naively they become an image recognition kind of problem.*



MACHINE LEARNING (DEEP LEARNING)

Major R&D field also in HEP.

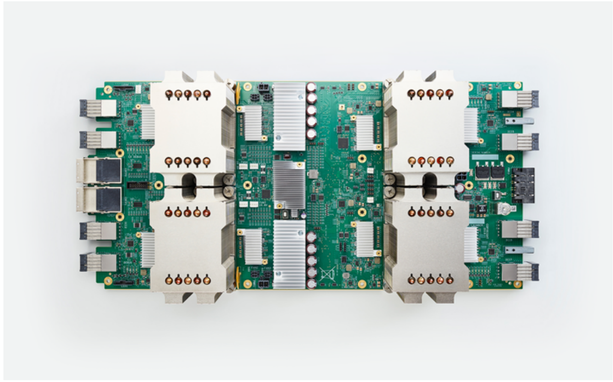
Tracking & pattern recognition: *naively they become an image recognition kind of problem.*

Fast simulation: *train a neural network to learn characteristic of a certain process and have it generate new examples based on such a training.*

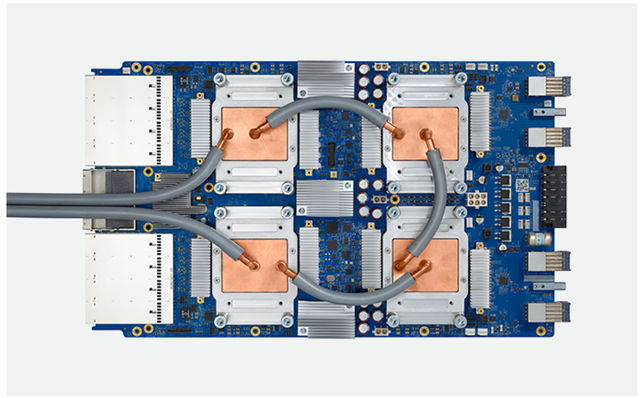
Typical approach in graphics programming is to cast a reduced set of light rays and then use a properly trained neural network to denoise and produce realistic shading.

TENSOR PROCESSING UNITS (TPUS)

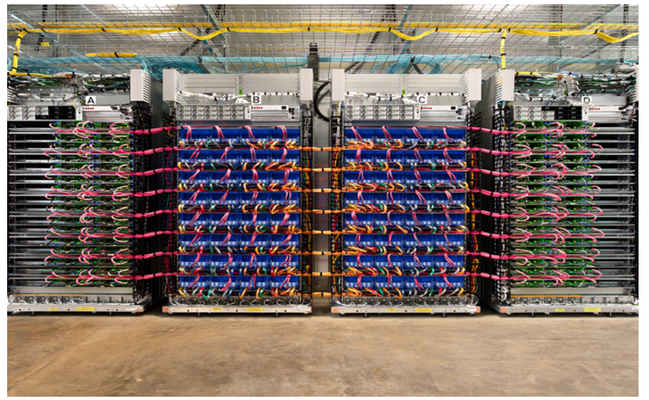
Performance and commercial importance of ML makes developing optimised silicon for ML a viable strategy.



Google's Cloud TPU v2
180 teraflops
64 GB High Bandwidth Memory (HBM)



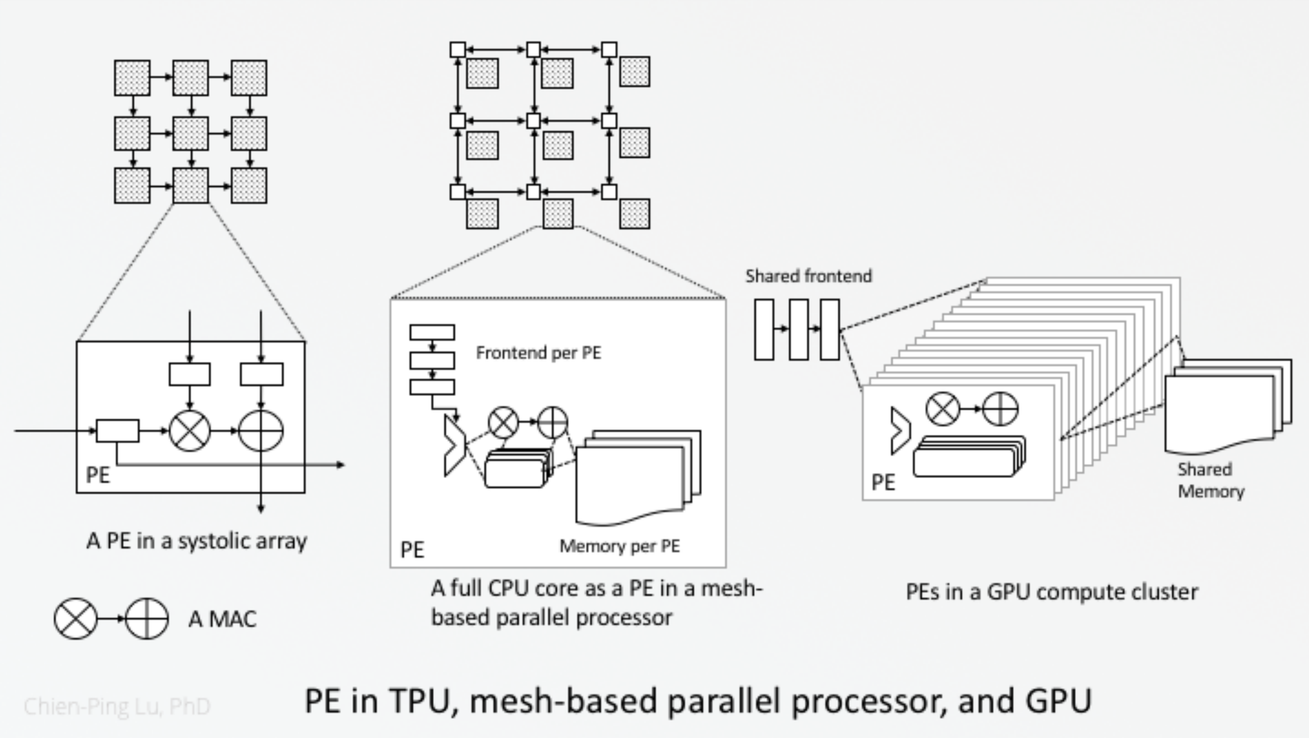
Google's Cloud TPU v3 Alpha
420 teraflops
128 GB (HBM)

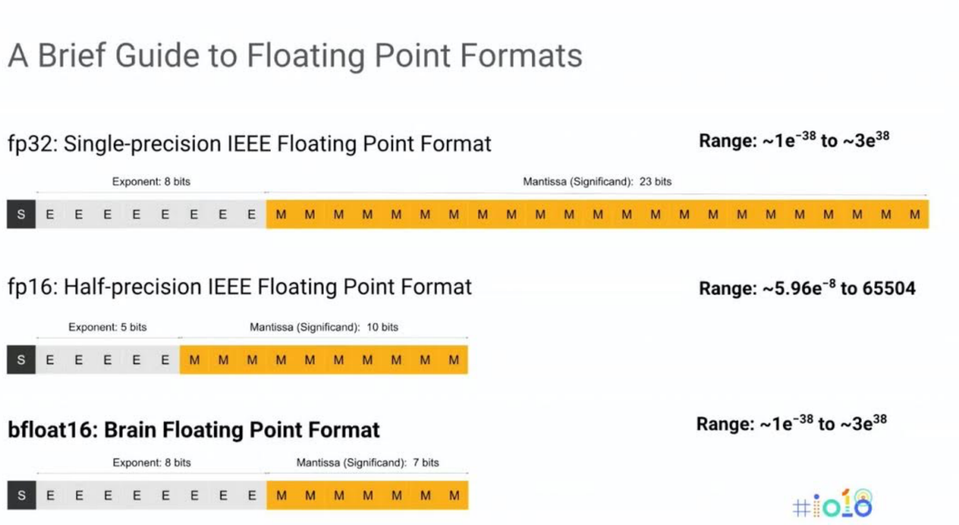


Google's Cloud TPU v2 Pod Alpha
aggregated 11.5 petaflops
2-D toroidal mesh network

When announced to the world (2016), Google claimed two orders of magnitude in sustained performance / watt to gain from this kind of architecture for Deep Learning kind of problems.

CPU VS GPU VS TPU





*TPU Hardware optimised for **low / mixed precision matrix multiplication, application of activation functions**. Tradeoffs on floating point representations to achieve wider range and on interconnect between processing elements (PE).*

GPU vendors (NVIDIA) are not sitting idle and started to provide hardware with similar tradeoffs (with almost comparable performance).

ML CHALLENGES

Concerns of 40 years ago still apply. *One thing is to find cats, another is to do full reconstruction of Pb - Pb collisions sampled at 50kHz for 22ms. Understanding systematics and transparency of the methodology are far from obvious.*

Huge, overhyped, field *with a lot of industry driven momentum behind. HEP is far from being the leader. Tools are mainly developed outside our world. Selection of tools viable for our 20+ years timescales and integration in our frameworks a challenge in itself. E.g.:*

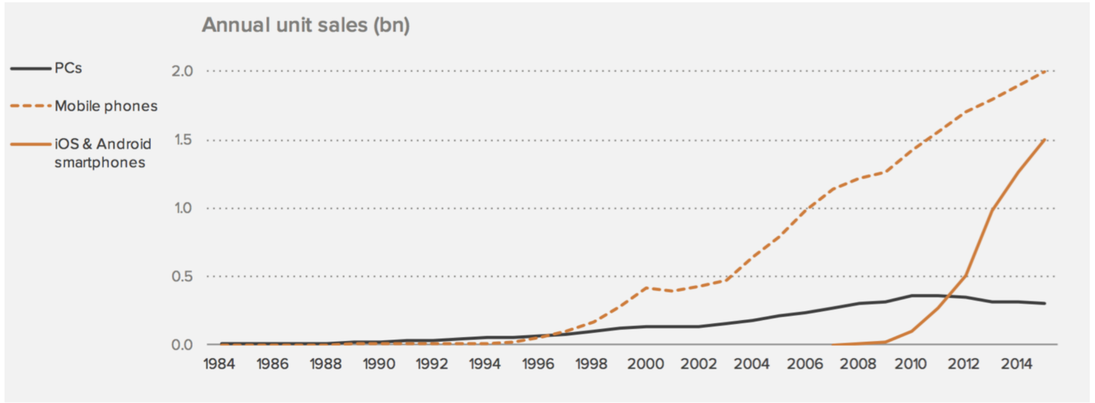
- *Reducing inference cost in terms of memory.*
- *Integrate nicely within experiment frameworks.*

Proprietary business models might be enforced on us: *Google provides TPU hardware only to workloads which run hosted on the Google Cloud, which poses obvious challenges for high data volumes like HEP's ones. Other service / hardware vendors might decide to go in a similar direction.*

REDUCED POWER USAGE: ALTERNATIVE (TO X86-64) ARCHITECTURES

Economy of scale

Over 7B mobile devices with low-power CPUs, mostly ARM based architectures. Can we use them in our data centers?



ARM

Leader in the mobile space, main challenger in the server market.

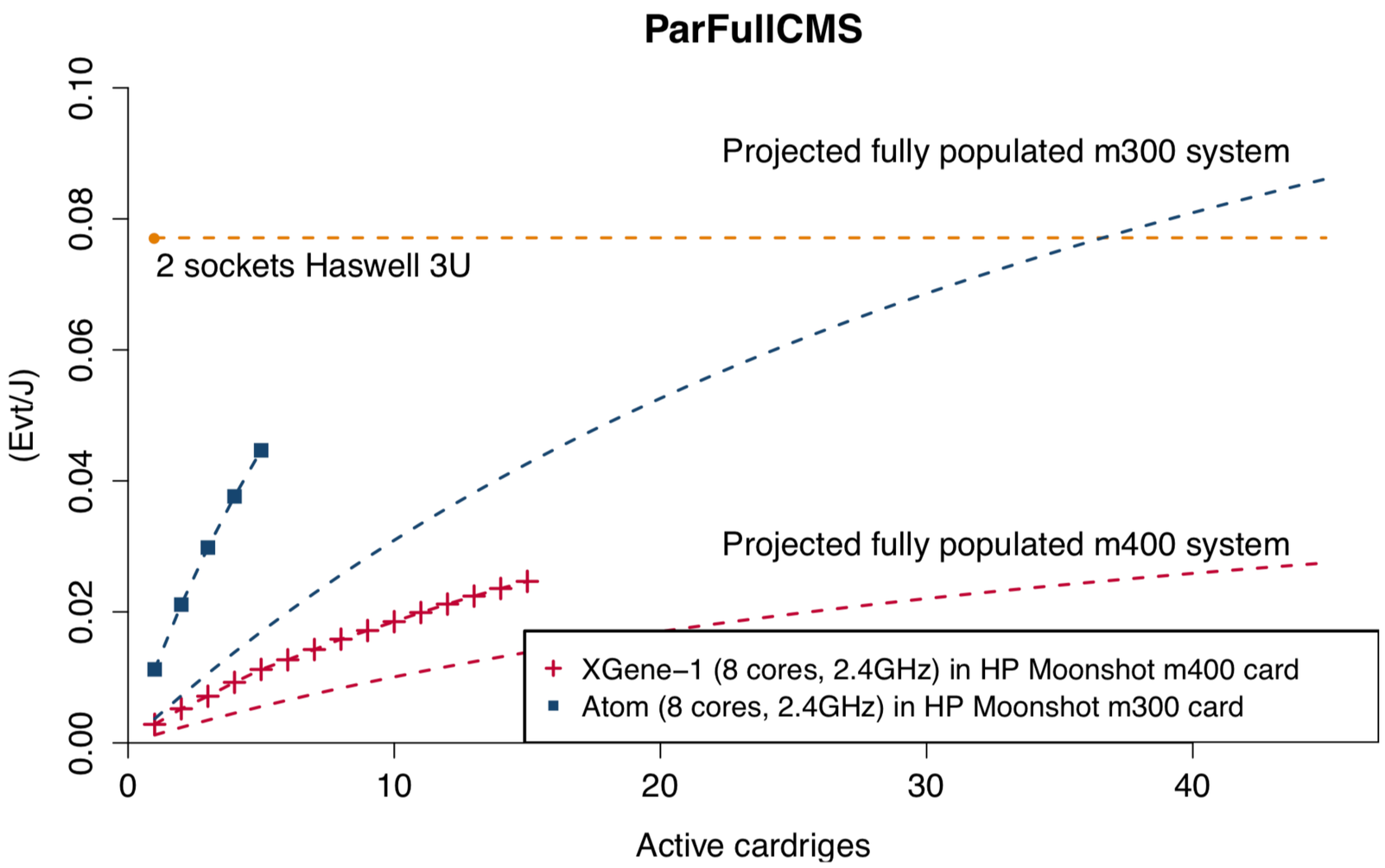
HEP attempts

Different architecture but still a general purpose CPU. *Close enough to make several* ***one to one*** *explorations possible:*

- *LHCb attempt: http://iopscience.iop.org/article/10.1088/1742-6596/513/5/052014/meta*
- *CMS attempt: https://arxiv.org/abs/1410.3441*
- *ATLAS attempt: http://inspirehep.net/record/1638522*

Augmented Reality experience and on-device Machine Learning are the driving factor for performance growth of mobile chips in the last couple of years.

ARM: MIXED EXPERIENCES SO FAR



Eulisse et al., 2015, doi:10.1088/1742-6596/664/9/092007

Promising tests...

Prototyping with embedded systems looked very promising...

...but not so easy.

Up to recent times, no real contender to Intel in the server market. CPUs are only part of the game. While quickly maturing, the ARM server ecosystem has been subpart until recently. Intel lead in manufacturing process a 1 - 2 generations advantage.

New scenarios?

Intel struggling with 10nm process and ARM server ecosystem improving over the last few years might give ARM vendors a window of opportunity in server market.

EXPLOITING PARALLELISM: FPGAS

Well known technology

Standalone FPGAs are already used throughout our detectors, triggers. E.g. ALICE uses it for hardware cluster finding in the TPC since Run 1.

Democratising FPGAs

New offering by Intel which integrate a Xeon CPU and a (formerly) Altera FPGA promises to increase flexibility of FPGA solutions in computing contexts and democratise their usage. Development environment allegedly more user friendly (OpenCL vs Verilog).

- Software triggers (see for example CMS R&D for CNNs on FPGA or LHCb R&D about mixed CPU / FPGA usage).
- Event simulation (see GeantV presentation at ACAT).

HETEROGENEOUS COMPUTING

Back to the future...

Compared to the rather homogeneous computing environments of the somewhat recent past (mostly x86 based HW) future of computing will probably be more variegated and specialised, like it was 30 / 40 years ago. HW accelerators (e.g. GPUs) will play a dominant role.

Integration between Online and Offline

Both ALICE and LHCb aim at online, in software, full reconstruction of Run 3 data, blending traditional roles of Online and Offline. Optimising the system as whole becomes a necessity.

Push for HPC resources

HPC resources are becoming extremely popular across the world. In the past they were severely limited by amount of RAM per core. If this remains the case we will need to think about ways to split our problems, especially for large event / timeframe sizes like ALICE in Run 3. E.g. multi-node reconstruction / simulation.

SUMMARY

Major challenges ahead...

Machine and detector upgrades will present major challenges both in terms of data throughput and in terms of the ability to reconstruct / analyse / simulate our detectors data.

...but also major optimisation opportunities!

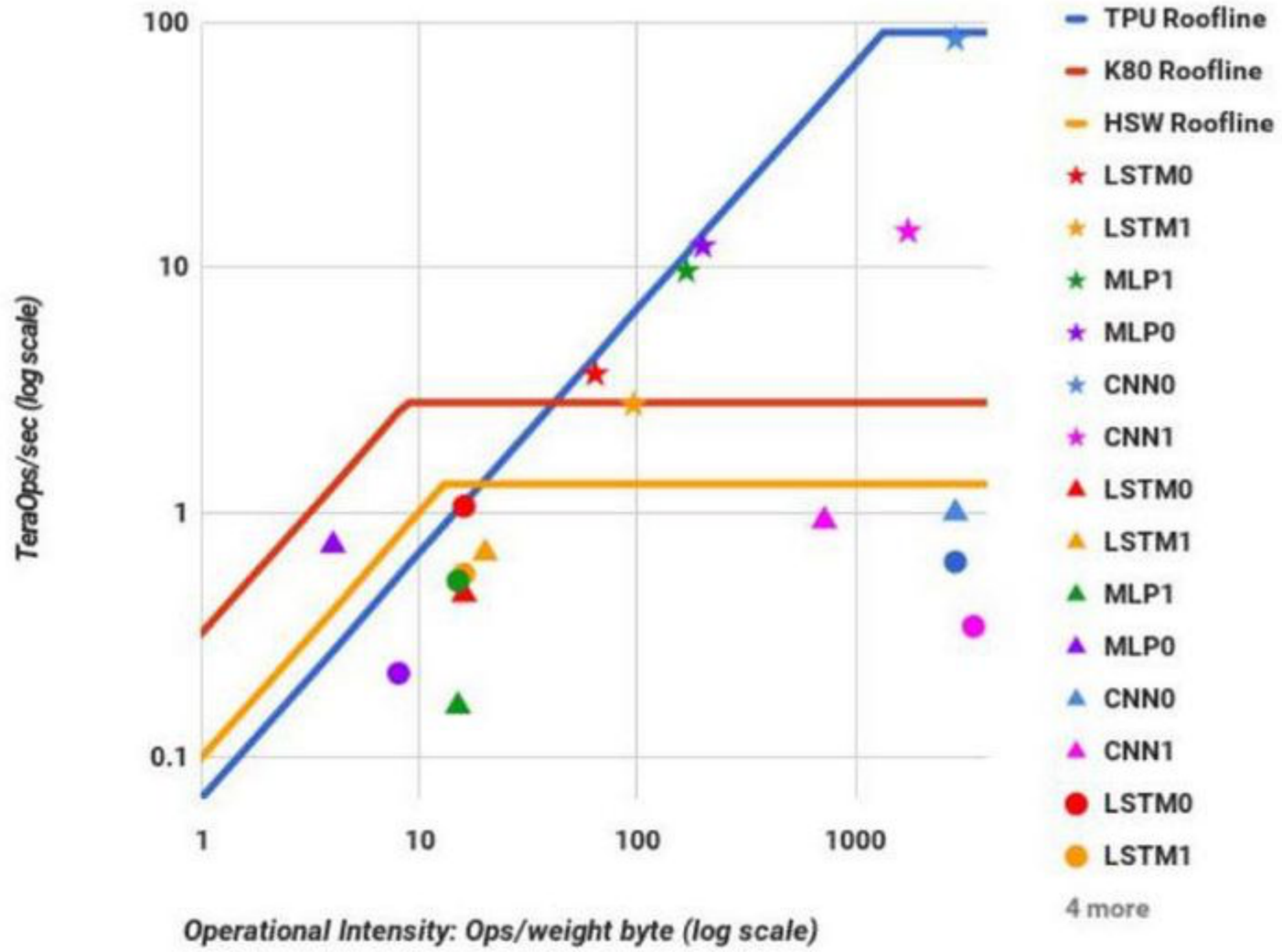
R&D on novel hardware and software solutions offer opportunities to match our future challenges (and possibly scale even further!).

No silver bullets...

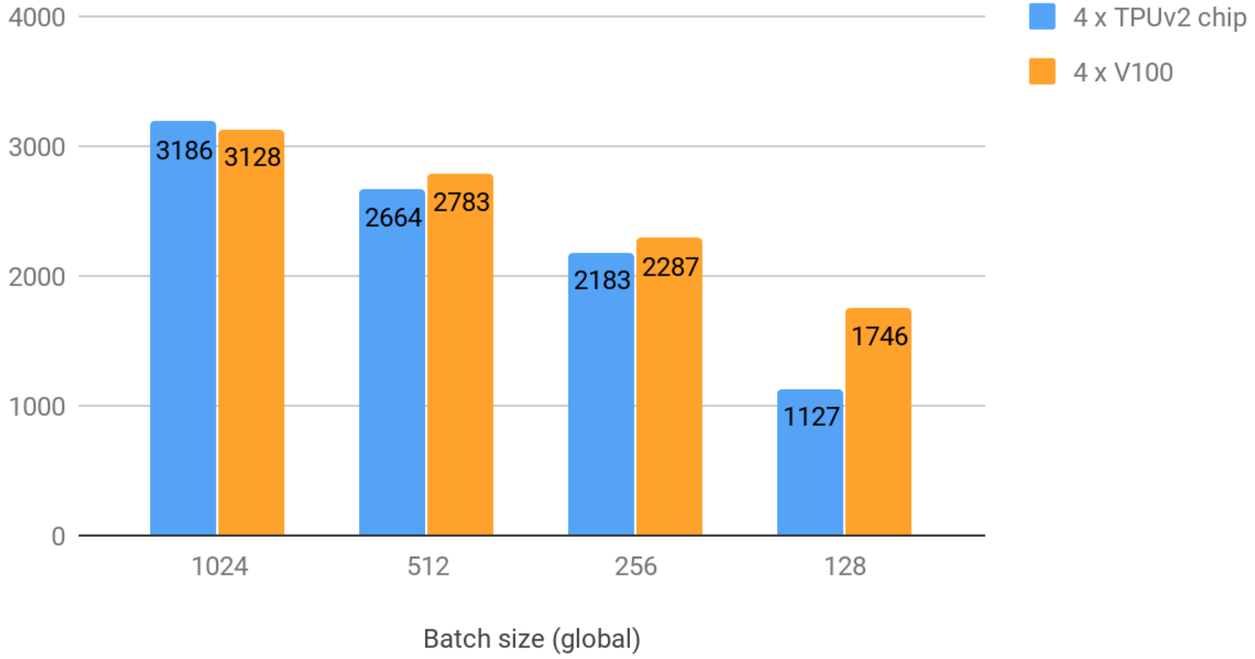
...but different solutions on multiple fronts. Flexibility to adapt to different computing environments is not only a divertissement to but a requirement to be able to process our data.

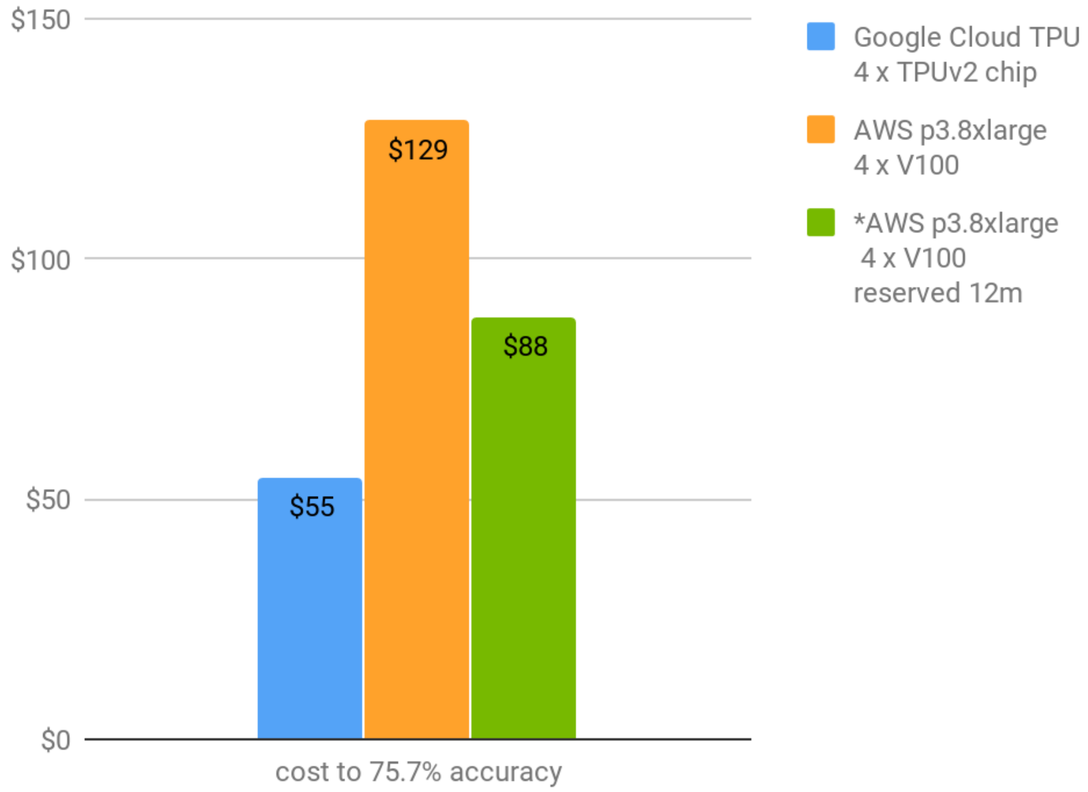
BACKUP

TPU VS GPU VS CPU: INTRODUCTORY COMPARISON



TPU VS GPU VS CPU: RECENT COMPARISON





https://blog.riseml.com/comparing-google-tpuv2-against-nvidia-v100-on-resnet-50-c2bbb6a51e5e