HOW 2019 Workshop

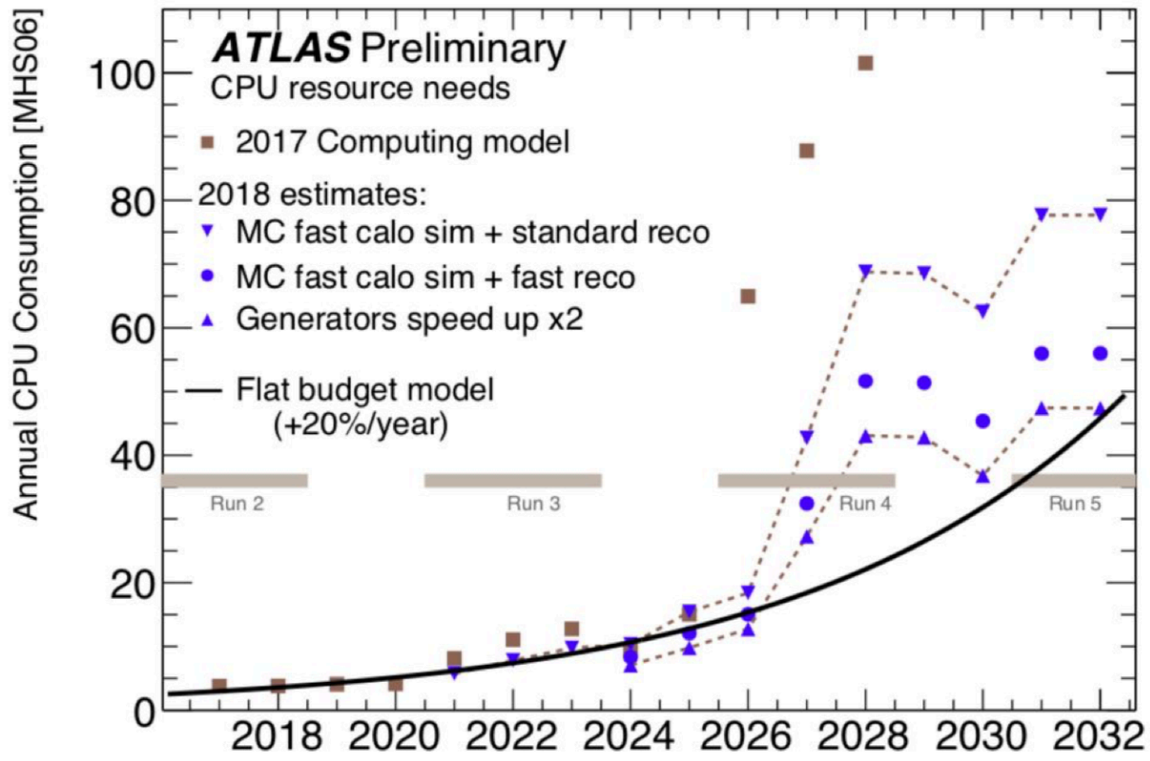
Stephan Hageboeck

Experiment Computing

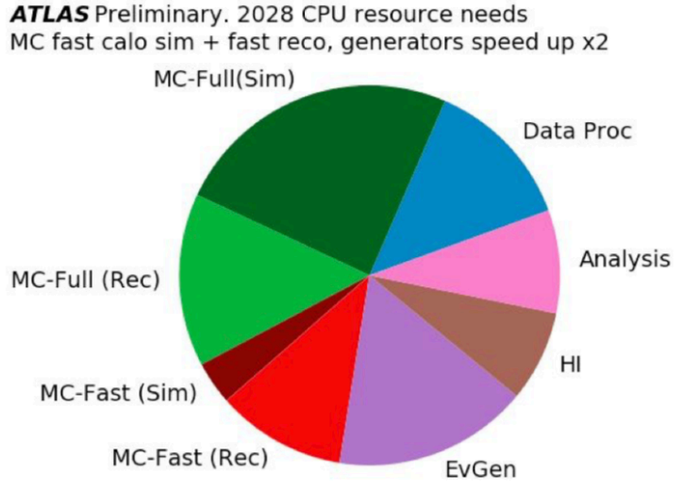
- LHC Run 4&5: Much larger datasets
  - Reconstruction increasingly demanding
  - Bottleneck: Simulation
  - “There is general consensus that the best performance/$$ will not be obtained with standard CPUs” CMS Talk

ATLAS Talk LHCb Talk

Fast vs Full simulation:
Run 3: 50% of simulation with fast sim
Run 4: 75% of simulation with fast sim



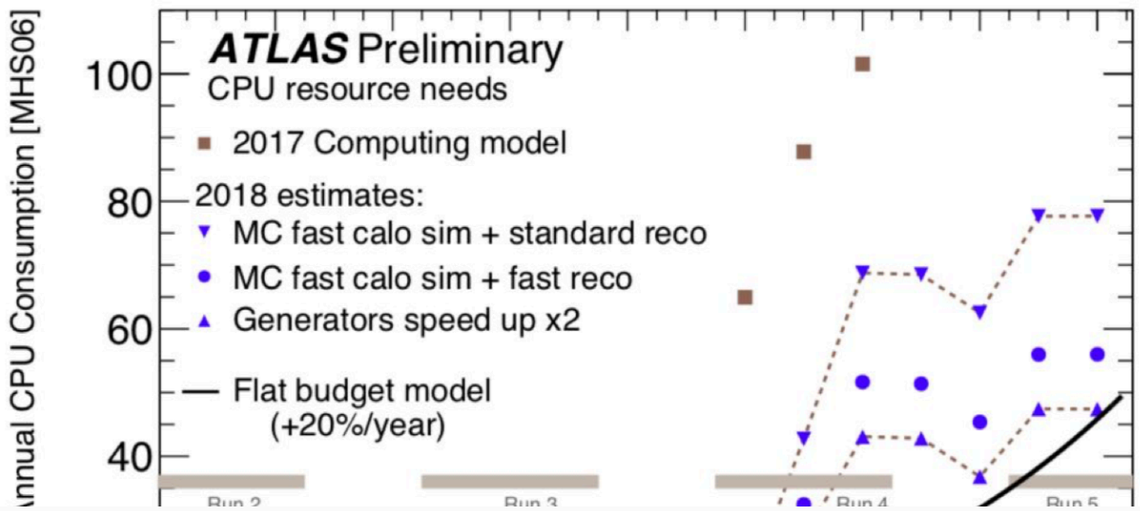
⅔ of global CPU time for simulation

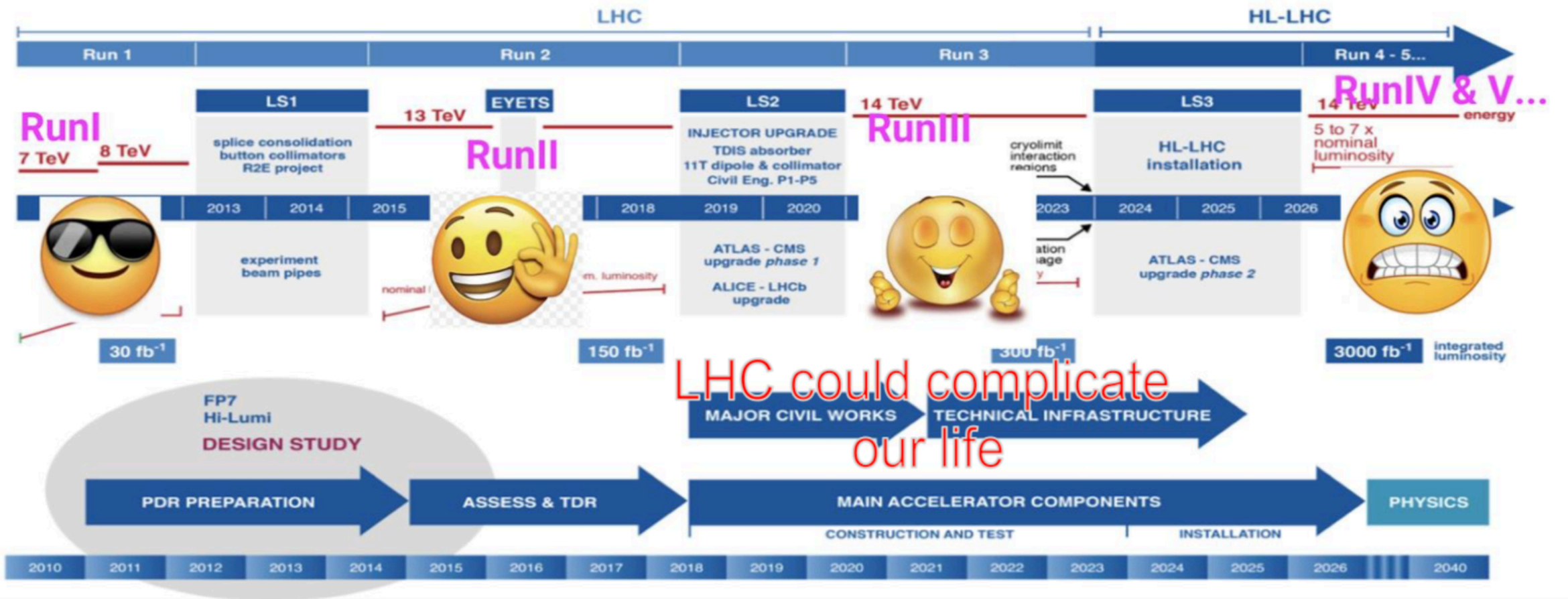


Experiment Computing

- LHC Run 4&5: Much larger datasets
  - Reconstruction increasingly demanding
  - Bottleneck: Simulation
  - “There is general consensus

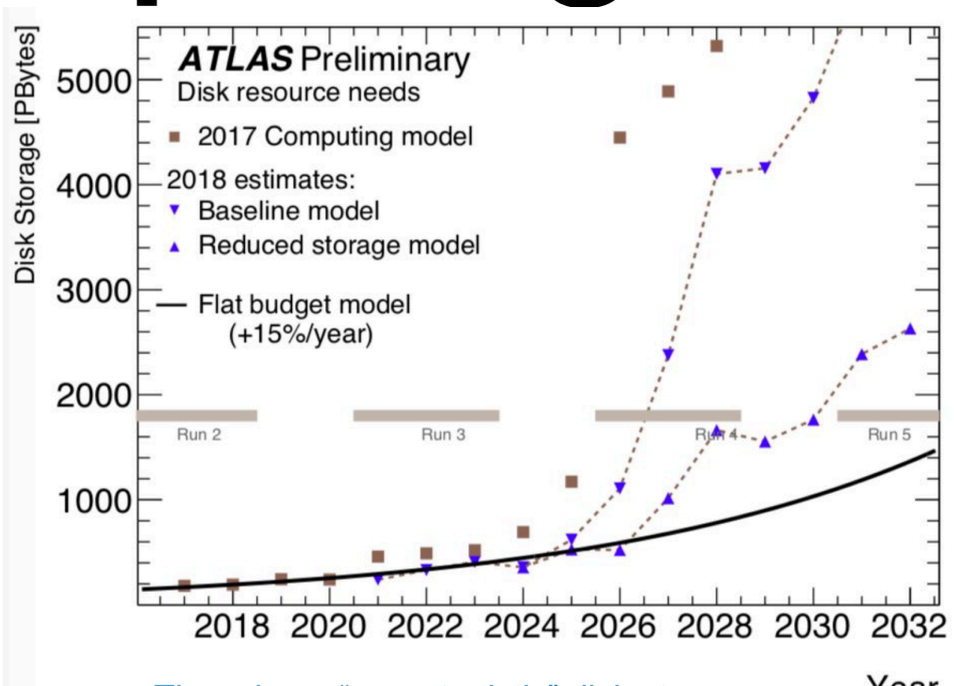
Fast vs Full simulation:
Run 3: 50% of simulation with fast sim
Run 4: 75% of simulation with fast sim



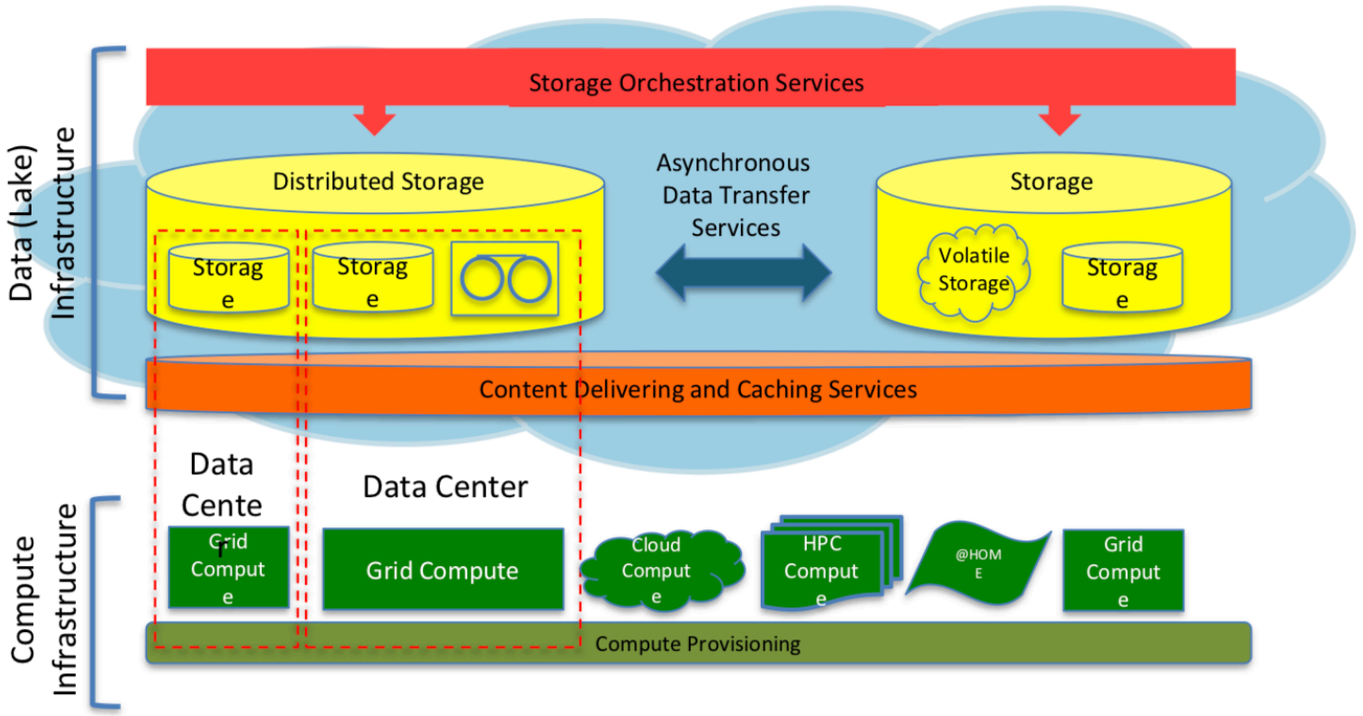


Experiment Computing

- Storage:
  - Never enough
  - Will have to rely on tape
  - Requires smarter data provisioning for analysers
  - Common tools / strategies (rucio, small pre-processed formats),
  - DOMA



- There is no “opportunistic” disk storage



Experiment Computing

- General directions seem to be
  - Trust in Moore's law & make algorithms faster
  - Shortcuts: "Fast" simulation (ML? GANs?)
  - Accelerators will become more important
    - ML Workflows: GPUs, TPUs (, FPGA)
    - What about the rest of the software? Can we help?
      See also Graeme's summary
- HPC to the rescue?
  - By Run 4: Enough FLOPs to process 30x current LHC data
  - Difficult to access: What technology/architecture? Intel/ AMD/IBM? GPUs? Data provisioning?
    HPC Talk I HPC Talk II

GPUs

GPUs in LHC experiments software frameworks

- Alice, O2
  - Tracking in TPC and ITS
  - Modern GPU can replace 40 CPU cores
- CMS, CMSSW
  - Demonstrated advantage of heterogeneous reconstruction from RAW to Pixel Vertices at the CMS HLT
  - ~10x both in speed-up and energy efficiency wrt full Xeon socket
  - Plans to run heterogeneous HLT during LHC Run3
- LHCb (online - standalone) Allen framework: HLT-1 reduces 5TB/s input to 130GB/s:
  - Track reconstruction, muon-id, two-tracks vertex/mass reconstruction
  - GPUs can be used to accelerate the entire HLT-1 from RAW data
  - Events too small, have to be batched: makes the integration in Gaudi difficult
- ATLAS
  - Prototype for HLT track seed-finding, calorimeter topological clustering and anti-kt jet reconstruction
  - No plans to deploy this in the trigger for Run 3

Accelerators and Memory

GPUs: Solution 1

GPUs - Programmability

- NVIDIA CUDA:
  - C++ based (supports C++14), de-facto standard
  - New hardware features available with no delay in the API
- OpenCL:
  - Can execute on CPUs, AMD GPUs and recently Intel FPGAs
  - Overpromised in the past, with scarce popularity
- Compiler directives: OpenMP/OpenACC
  - Latest GCC and LLVM include support for CUDA backend
- AMD HIP: **Solution 1?**
  - Interfaces to both CUDA and AMD MIOpen, still supports only a subset of the CUDA features
- GPU-enabled frameworks to hide complexity (Tensorflow)
- Issue is performance portability and code duplication

24

GPUs: Solution 2?

ALPAKA — DOUBLE PRECISION Y = A * X + Y

```
struct DaxpyKernel
{
    template< typename T_Acc >
    ALPAKA_FN_ACC void operator()(
        T_Acc const & acc,
        double const & alpha,
        double const * const X,
        double * const Y,
        int const & numElements
    ) const
    {
        using alpaka;
        auto const globalIdx = idx::getIdx< Grid, Threads >( acc )[0u];
        auto const elemCount = workdiv::getWorkDiv< Thread, Elems >( acc )[0u];
        auto const begin = globalIdx * elemCount;
        auto const end   = min( begin + elemCount, numElements );
        for( TSize i = begin; i < end; i++ )
            Y[i] = alpha * X[i] + Y[i];
    }
};
```

Laser Acceleration / ALPAKA (ACAT)

Use CMAKE to change Accelerators or common header

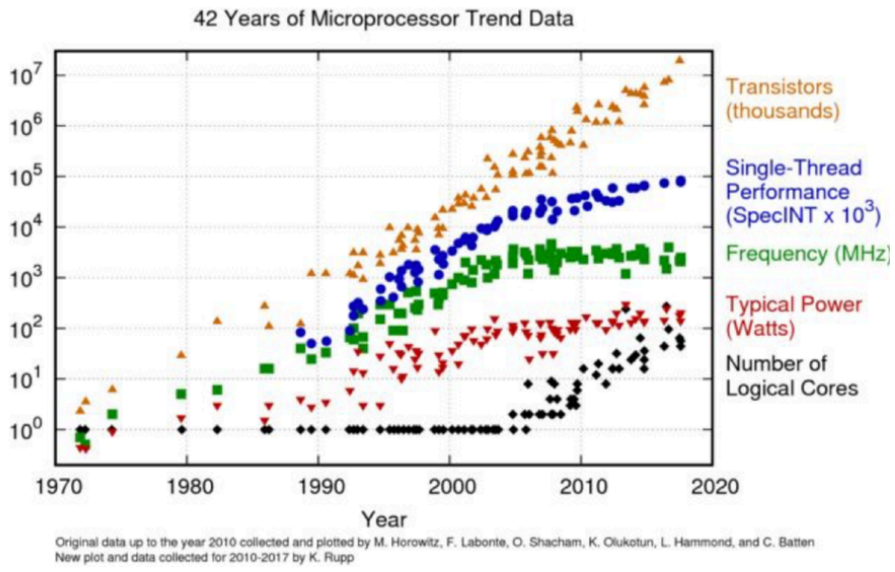
Other Experiments

- Belle, Ice cube, Virgo/Ligo, Dark Matter, LSST, Dune, …
  Agenda
  Recommend: Dark matter overview
  (Questionnaire for different DM experiments. Very diverse.)

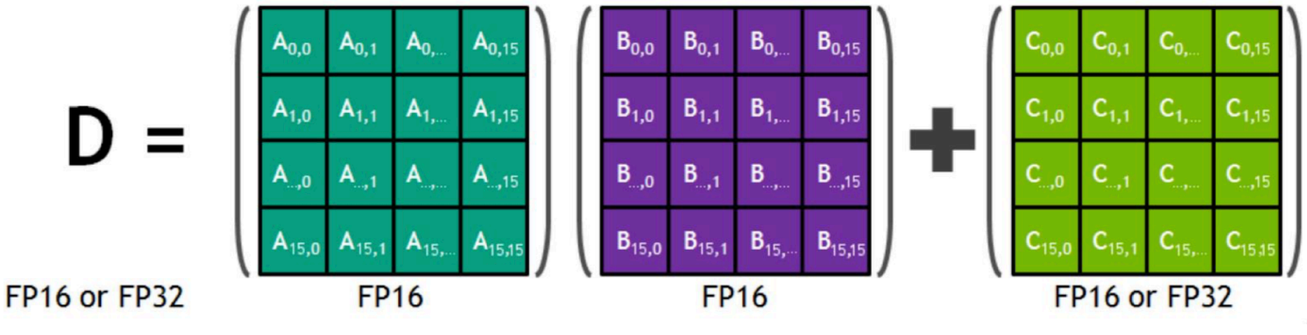
- No Run 4/5 problem, but similar questions
  - More ML
  - How to make use of accelerators? HPC?
  - How to store & distribute data?
  - Grid & batch seem to be workhorses

Hardware Watch

- CPU Architectures & Accelerators
  - Moore's law seems to hold (almost) in CPUs & GPUs, AMD is back in CPU server market
  - GPUs with tensor cores, TPUs:
    - Only FP16
    - Very fast & high FLOPs/W
  - Is CUDA really the solution?
    - De-facto standard, but not portable

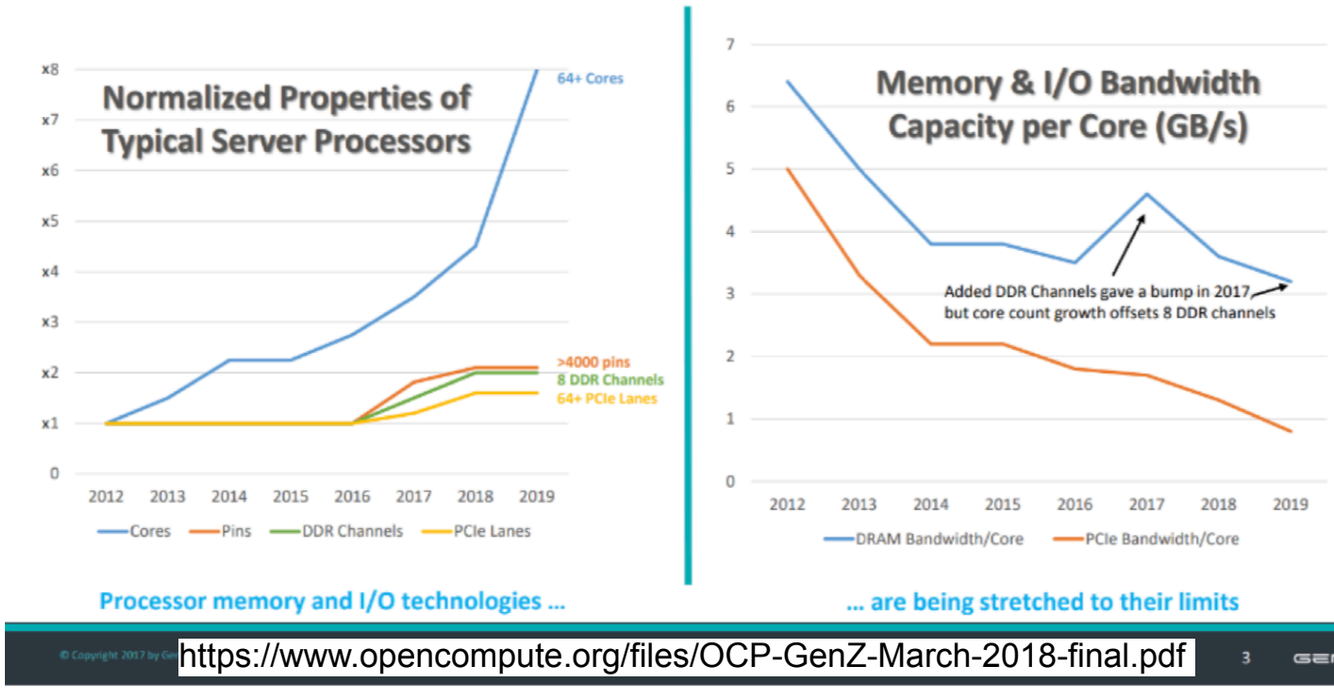


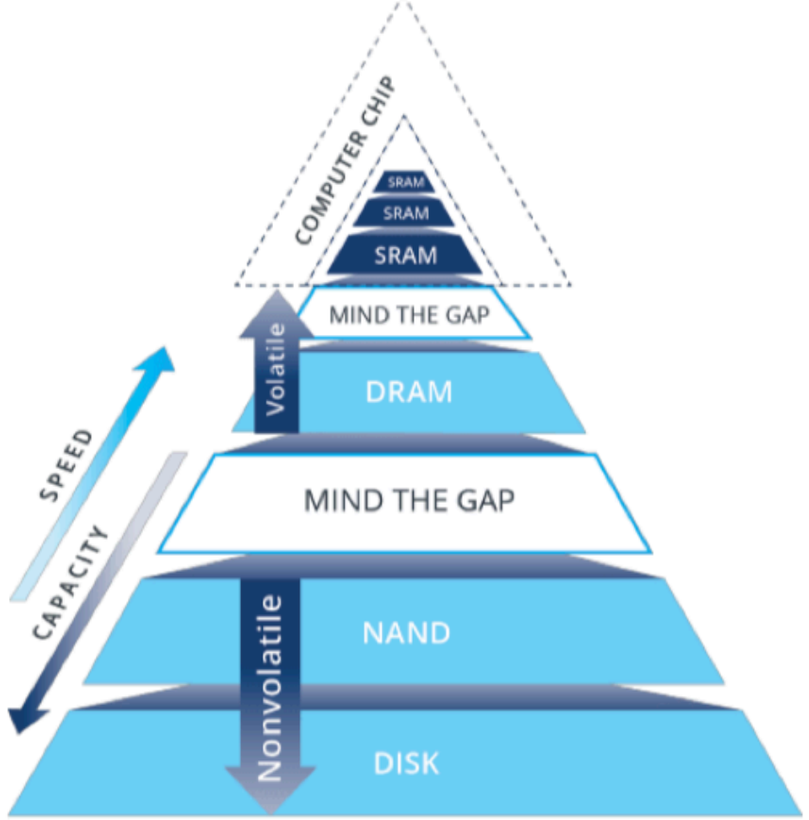
Feature	Volta (V100)
Process	12nm
CUDA cores	yes
Tensor cores	yes
RT cores	NA
FP performance	FP16: 28 TFLOPS FP32: 14 TFLOPS FP64: 7 TFLOPS Tensor: 112 TFLOPS



Memory

- Significant gap between SRAM caches and DRAM
  - Latency unchanged
  - Bandwidth/core falls
- Second gap to persistent memory





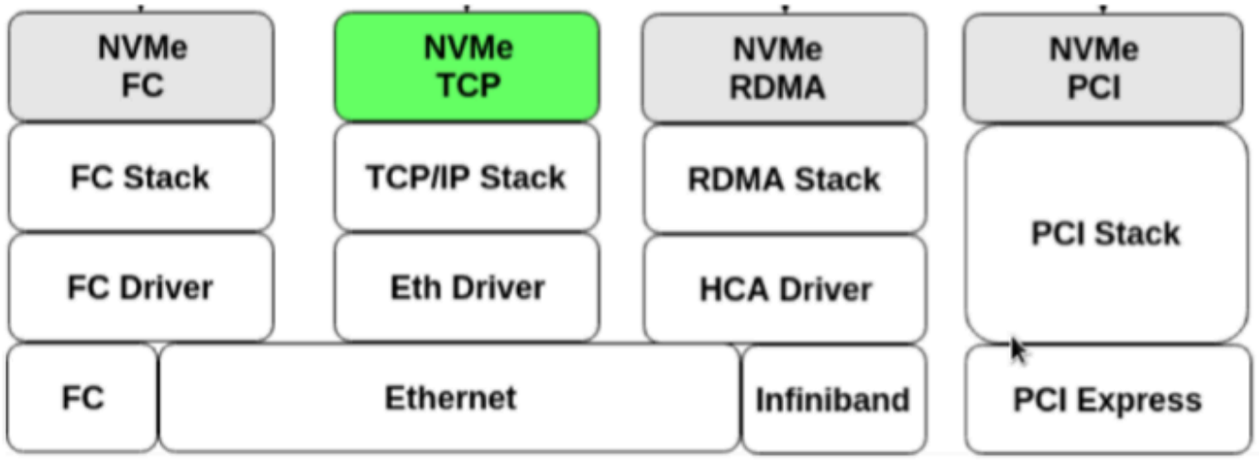
https://www.eetimes.com/author.asp?section_id=36&doc_id=1334088#

Non-Volatile Memory

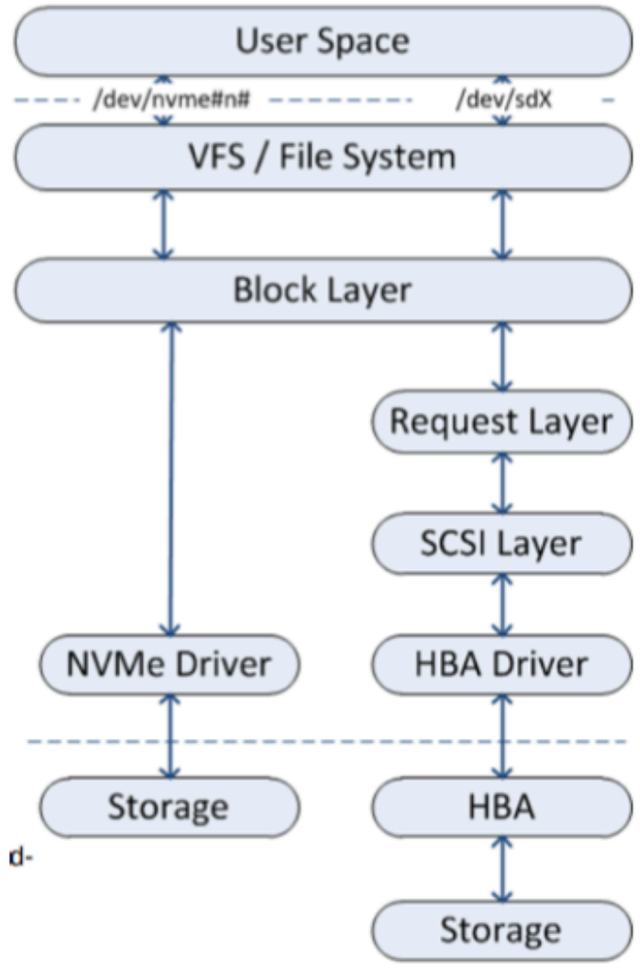
Solid-State Storage (II) - NVMe/NVMe-oF

NVMe and NVMe over Fabrics (NVMe-oF) is the center of industry attention and activity

1. NVMe (NVM express) eliminates multiple software layers in the OS stack.
2. NVMe-oF extends NVMe interface to other interconnects (PCI-e, IB, FC, DC Ethernet, …)
3. NVMe being expanded (e.g., enclosure management, multi-path, device management, …)
4. Aiming to be "lingua franca" for high performance solid state storage - unleash SSD potential
5. Allows for more radical solid state storage architectures/systems.



Source: Kam Eshghi (Lightbits Labs)



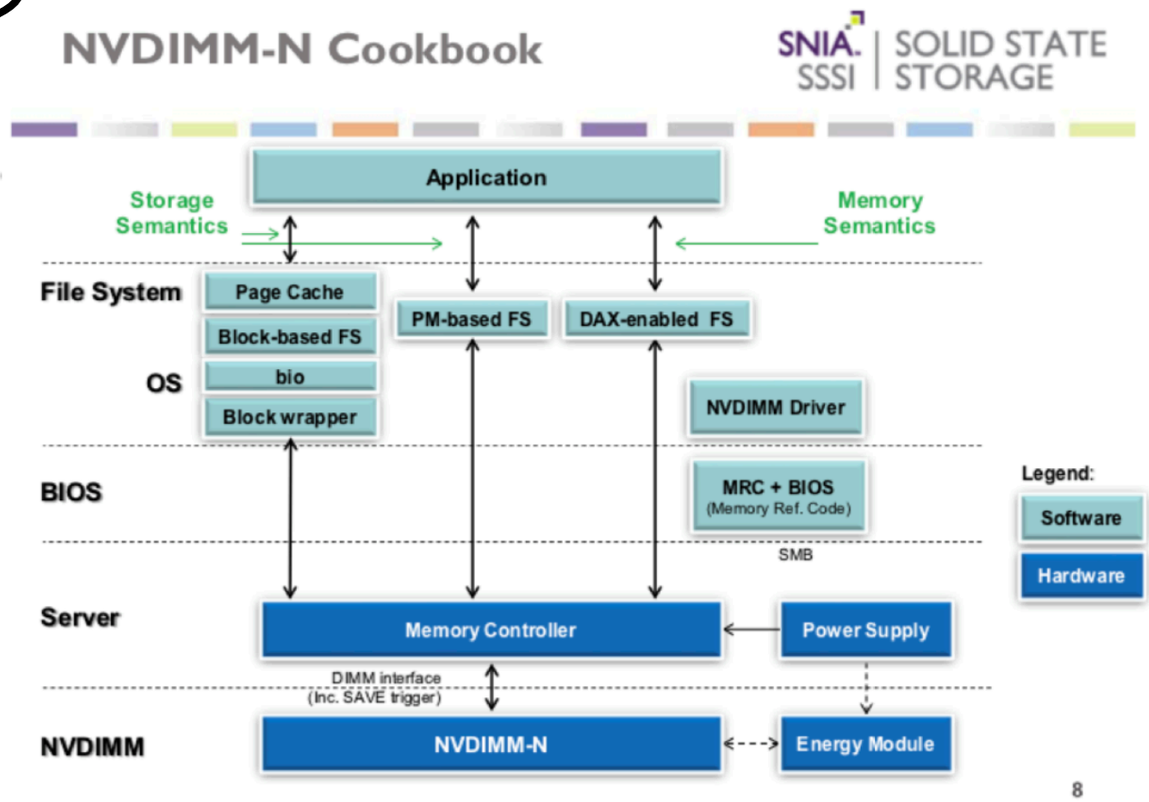
Source: K. Bush, Intel, 2014 Flash Memory Summit

Solid-State Storage (III) - NVDIMM

Persistent Memory (NVDIMM)

- Non volatile memory on the CPU memory bus (DDR4/DDR5)
- DDR4/DDR5 DIMM physical form factor
- CPU and system support required.
- Higher density and lower costs than DRAM is expected
- DRAM-like access latencies are expected (~ x3 higher)
- Designed for “ultimate” I/O performance.
- Programming models and usage are hot topics in academia and industry - high market expectations
- Enables new computing models which could benefit caching, DAQ, burst buffer, in-memory DB,
- 3D NAND flash (with hybrid DRAM) and 3D XPoint memory expected to be memory technology of choice for NVDIMM.

Hardware and software components needed for Persistent Memory are becoming available.
NVDIMMoF in discussion (what we had with SGIs NUMA machines years ago)



Memory Technologies

Summary

- Computing challenges ahead: Software needs to cope with increasing amounts of data & new hardware

- Hardware
  - Accelerators seem to be inevitable
  - Try not to bet on only one
  - Try to identify the best (hardware-agnostic?) library

- Might take a bit leap in terms of I/O performance & memory latency soon

- Collaboration
  - Increasing awareness that similar problems ahead
  - That’s what HSF is for