STORAGE DEVELOPER CONFERENCE
SDC 22
Fremont, CA
September 12-15, 2022
BY Developers FOR Developers
A SNIA Event

HPC Scientific Simulation Computational Storage Saga

09/2022

LA-UR-22-28008

Gary Grider
HPC Division
Los Alamos National Laboratory

History: Eight Decades of Production Weapons Computing to Keep the Nation Safe

Maniac

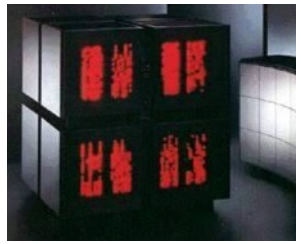
IBM Stretch

CDC

Cray 1

Cray X/Y

CM-2

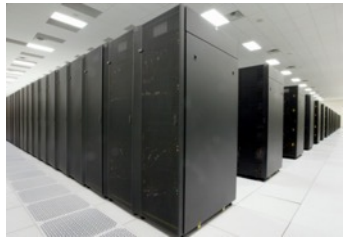


CM-5

SGI Blue Mountain

DEC/HP Q

IBM Cell Roadrunner



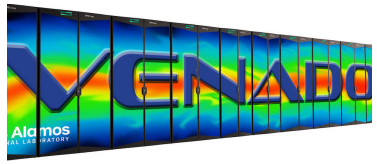
Cray XE Cielo

Cray Intel KNL Trinity

Ising DWave

Cross Roads

Venado



Background: HPC Scientific Simulation Systems

Trinity – circa 2016

- Haswell and KNL
- 20,000 Nodes
- Few Million Cores
- 2 PByte DRAM
- 4 PByte NAND Burst Buffer ~ 4 TByte/sec
- 100 Pbyte Scratch PMR Disk File system ~1.2 TByte/sec
- 60PByte/year Sitewide Campaign Store ~ 50 GByte/sec
- 60 PByte Sitewide Parallel Tape Archive ~ 3 Gbyte/sec

Circa 2023

- 10 PB DRAM
- 100 PB Flash
- Half Exabyte spinning disk

I know its not Tier1 sized but at LANL its for one job for several years.
10 PB files and 200 PB Campaigns
For a single user/small user team

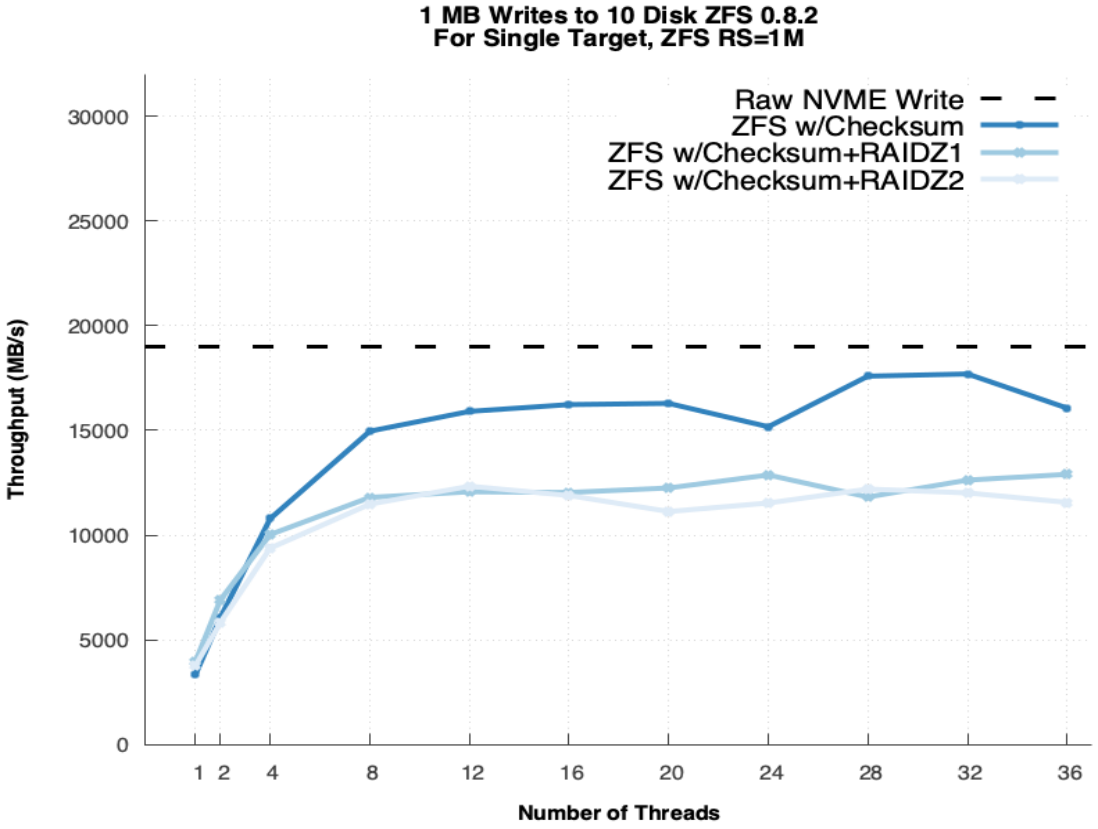
Topics: Crawl, Walk, Run - with much help from our partners!

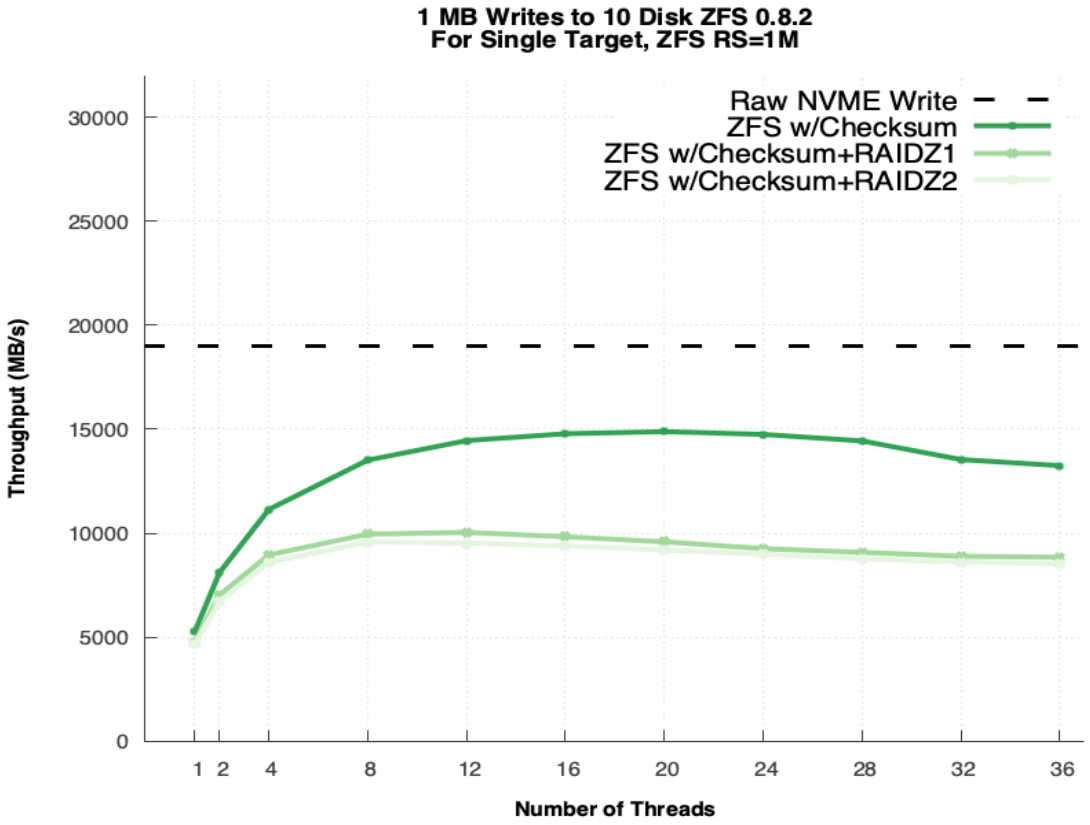
- ABOF 1.0 (Eideticom, Aeon, Nvidia, SK hynix)
  - Format agnostic operations (compression, erasure, encoding)
- DeltaFS->Ordered KV-CSD (CMU and SK hynix)
  - Format aware, record-oriented applications with a single-dimension, easily shard-able indexing
- ABOF 2.0 plans (Eideticom, Aeon, Nvidia, SK hynix, others?)
  - Format aware, column-oriented applications, multi-dimension, difficult to shard indexing

ABOF 1.0 (Eideticom, Aeon, Nvidia, SK hynix) Format agnostic operations: compression, erasure, encoding

Memory Bandwidth Intensive Offloads

Why Offload? ZFS Checksums, Erasure, Compressive Server memory bandwidth is problematic and expensive

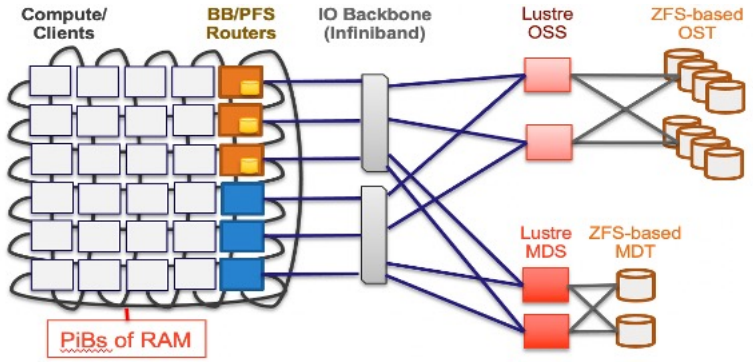




- Intel Platinum (Dual Socket)
- AMD EPYC (2nd Gen)

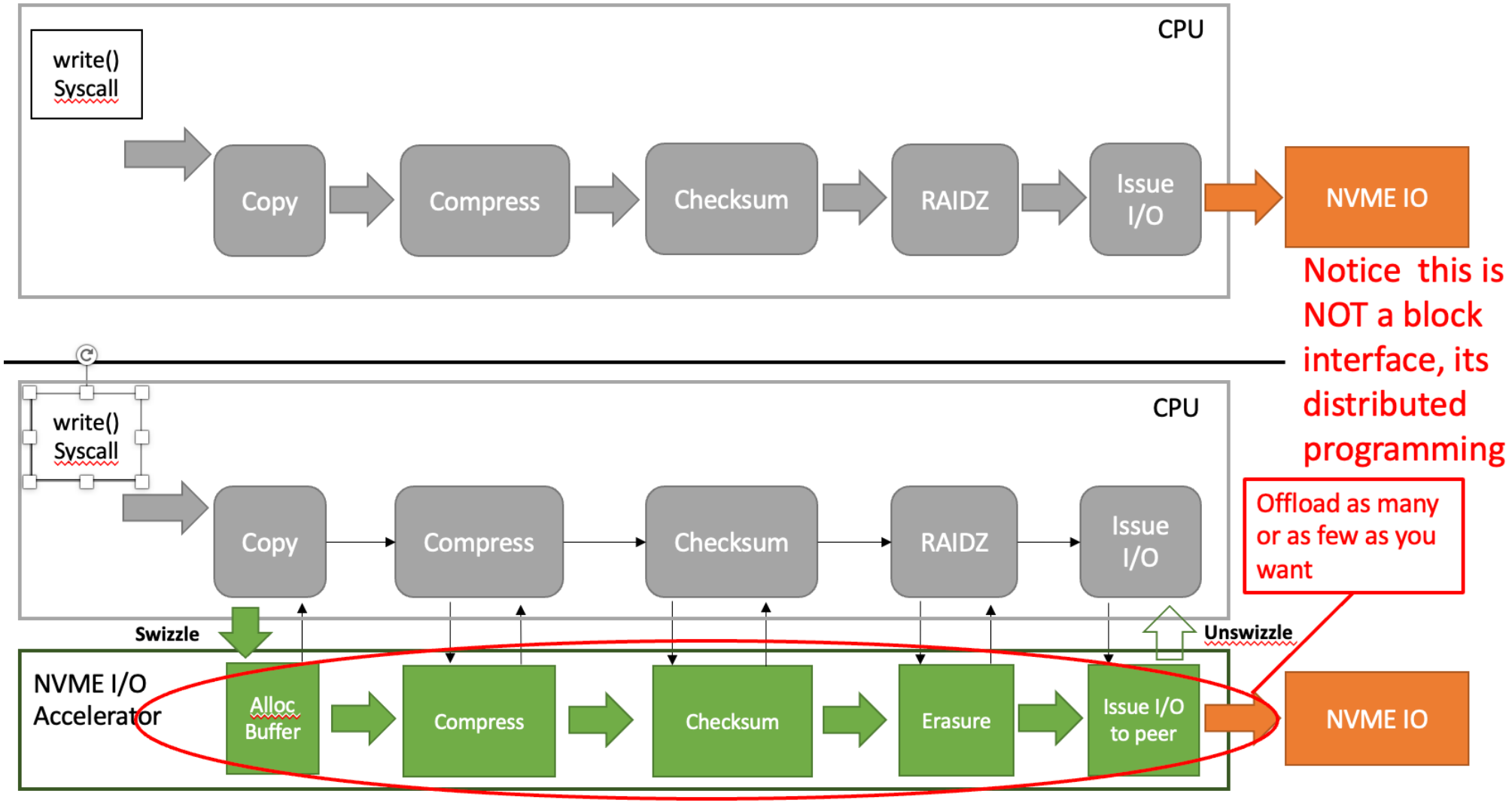
How to Consume: File System Services Offload

- Requirements
  - Require parallel file system support
  - Millions of processes writing/reading single parallel and multiple files
  - Under Lustre we can use ZFS as the component file system
  - Flexibility in where we run what (erasure, encoding, compression/decompression) in cpu (kernel) running ZFS, in BOF Nic in BOF CSA/CSD/CSP to match changing performance economics
  - Attempt to follow emerging NVME CS standard (NVME using peer to peer etc.)
  - Solution needs to be broadly available (We chose ZFS)



- Computational Storage Benefits/Opportunities
  - Increase compression rates from 1.06:1 -> 1.3:1 for scientific data
  - Enable expensive coding/decoding to protect against correlated failures
  - Higher per-server and per-device bandwidths
  - Lower server costs and quantities
  - Enable more than block as only interface (distributed heterogeneous computing)

Notional fixed function offloads in ZFS



Accelerated Box of Flash: Powerful Computational Storage for Big Data Projects

Radically new approach to storage acceleration aids data manipulation for research and discovery

MARCH 21, 2022

Recent Press Release

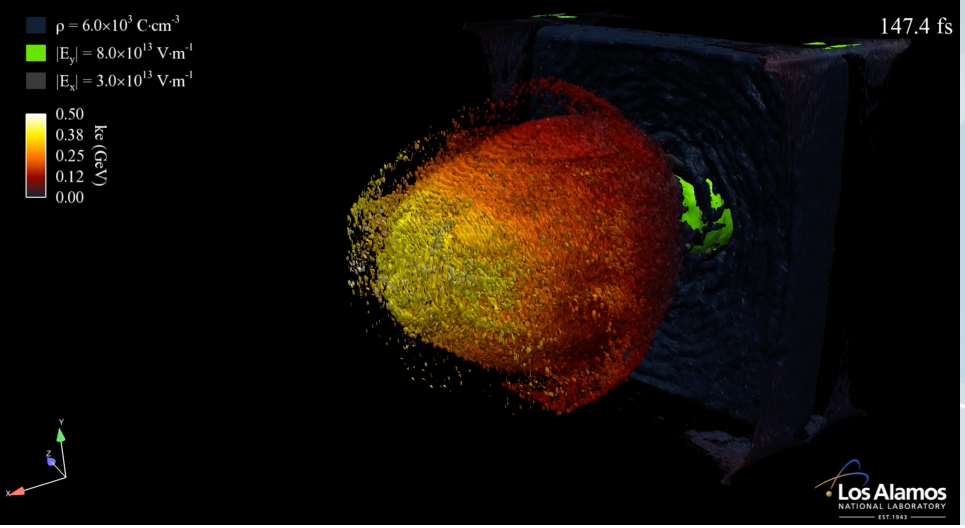
An excellent partnership!

Partnership to demonstrate NVME Computational Storage based ZFS accelerated ABOF (distributed popular host kernel based FS app distributing functions to ABOF smart nic and NVME accelerator)

Eideticom – NoLoad™ software
AEON – enclosure hardware design/engineering
NVIDIA – Bluefield 2 technology
SK hynix – Flash Storage
LANL – ZFS, ZFS offload interface, Linux Kernel offload layer

https://discover.lanl.gov/news/0321-computational-storage

Hopefully will help guide NVME Computational Emerging Standards.



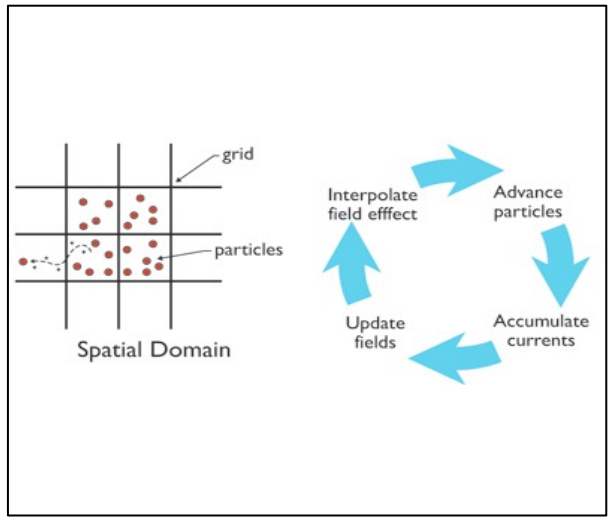
Plasma Studies

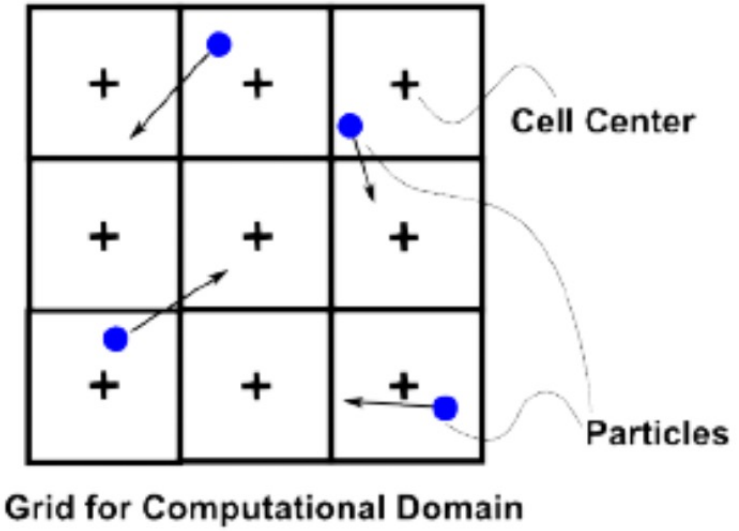
DeltaFS->Ordered KV-CSD (CMU and SK hynix)

Format aware, record-oriented applications with a single-dimension, easily shard-able indexing

Vector Particle in Cell (VPIC) our Record Based/Single Dimensional Index Application

- Particle-in-cell MPI code (scales to ~100K processes)
  - Fixed mesh range assigned to each process
  - Record: 32 – 64 Byte particles (id, cell id, energy, …)
  - Particles move frequently between processes
  - Million particles per node (Trillions of particles in target simulation)
  - Interesting particles identified at simulation end (say 1000 interesting particles)
- Why offload?
  - Retrievals are 9 orders of magnitude smaller than total data



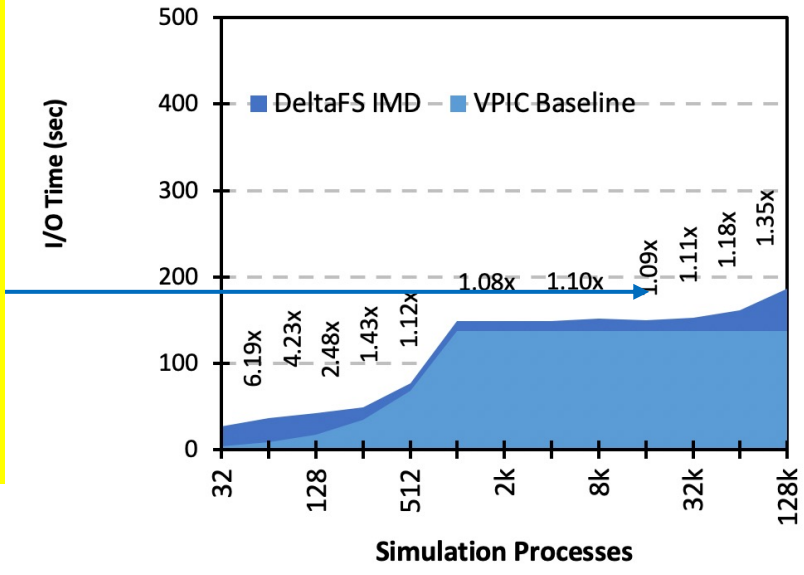


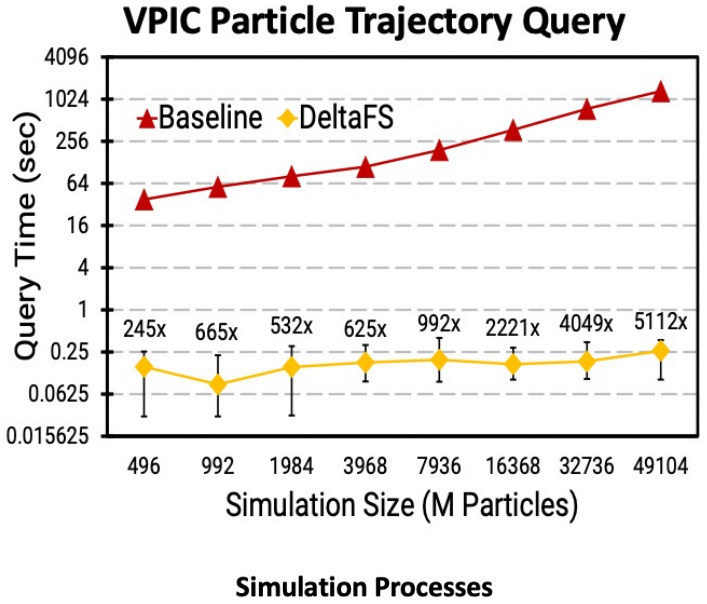
DeltaFS - Near-device Indexing and Analytics

- Requirements
  - Simulations run under intense memory pressure (app may use 90%)
- Computational Storage Benefits/Opportunities
  - Speedups for post-hoc analysis (1000x speedup demonstrated)
  - Less reliance on massive compute tier as a large merge sort space

Get efficiency and lower time to solution (1000X)

Add a little time indexing on the way out and get 1000X on analysis step (the indexing must scale and be efficient (perfect offload opportunity)

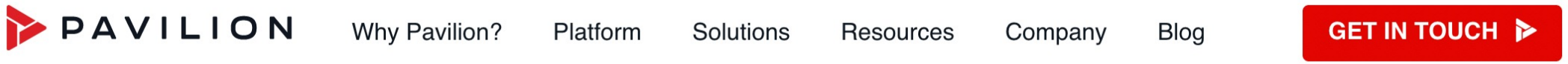




Application thought it was writing/reading from 1 file per trillion particles, really writing records to massive parallel distributed sharded by particleID KVS!
8 Billion Particle Ops/Sec.
(yes Billion)

(papers at PDSW 15, PDSW 17, SC19 (Best Student Paper)

KV-CSA Key Value Computational Storage Array



FEATURED

Los Alamos National Laboratory and Pavilion Partner to Explore Analytics Offloads to Computational Storage Arrays

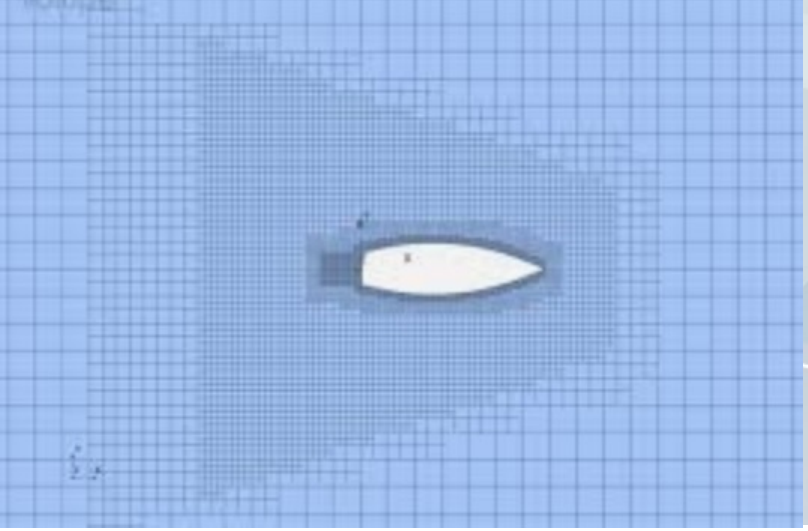
05.25.2022

Full Ordered Key Value Store offloaded to an NVMEOF Storage Array
Excellent performance
Extensions on SNIA KV on NVME
Please ask for our semantic extensions (think mput, mget, explicitly controlled compaction)

SK hynix Ordered KV-CSD Prototype Revealed at FMS ‘22

- SK hynix – LANL collaboration
- Fully offloaded ordered key value with point and range query capability (put, get, mput, mget, etc.)
- Offloaded all the way to hardware to the ZNS device level
- Extensions – control of compaction, and more
- Excellent performance

Full Ordered Key Value Store offloaded to an NVMEOF Storage Array

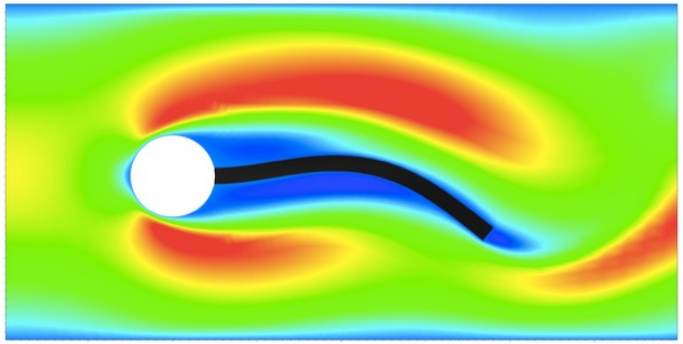


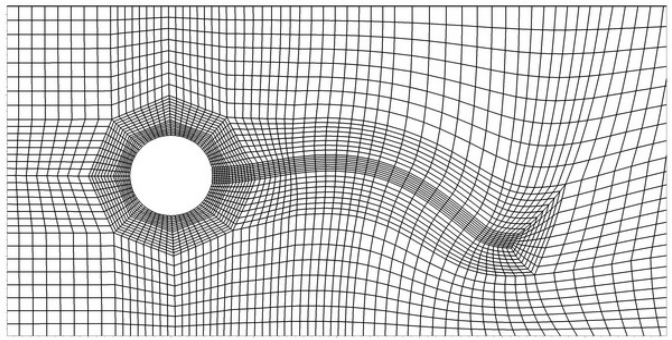
Grid based methods
Adaptive Mesh
Refinement
Finite
Element/Difference,
Hydro, Transport

ABOF 2.0 plans (Eideticom, Aeon, Nvidia, SK hynix, others?)

Format aware, column-oriented applications, multi-dimension, difficult to shard indexing

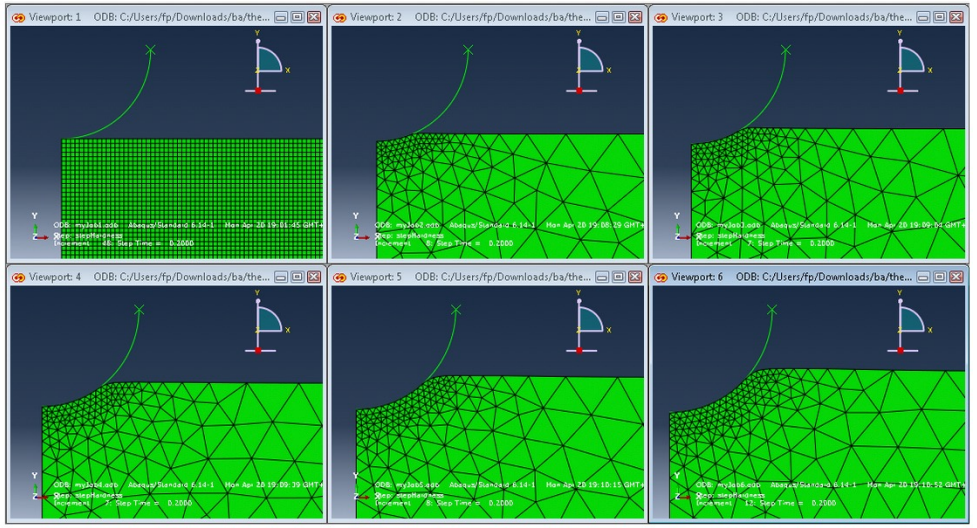
What's a Grid Method and an Adaptive Mesh Refinement (AMR)?



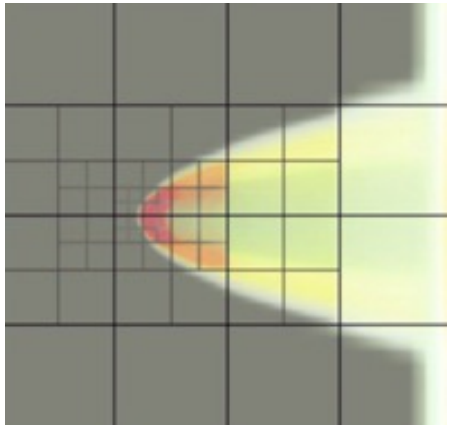


ALE – Advanced Lagrangian Eulerian

http://web.cs.ucdavis.edu/~ma/VolVis/amr_mesh.jpg



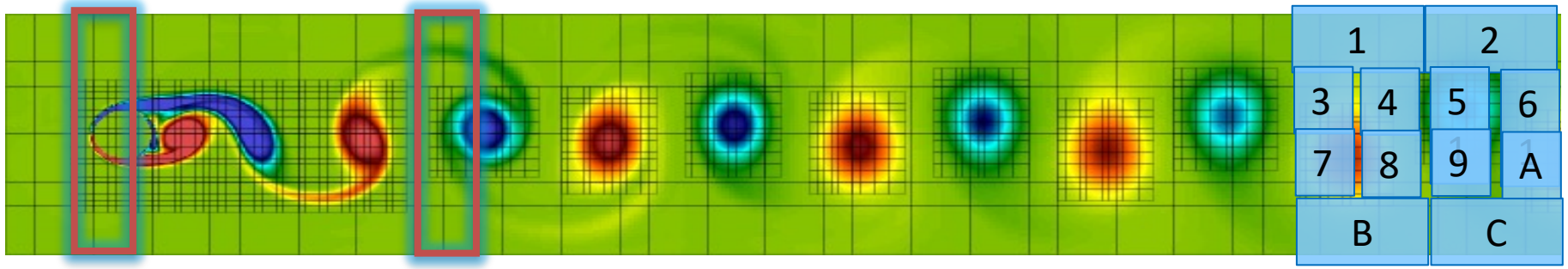
AMR



Eulerian AMR

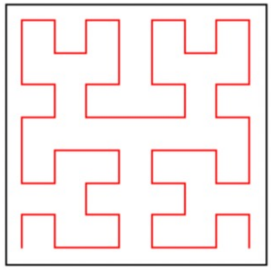
- Lagrangian (mesh deforms)
- Eulerian (mesh doesn't deform)
- AMR – mesh adapts (refines where the action is)
- Why? – to fit a problem that is way to big for your RAM
- AMR eliminates compression, copy on write, other low hanging fruit

Indexing Multi-Dimensional Unstructured Adaptive Meshes

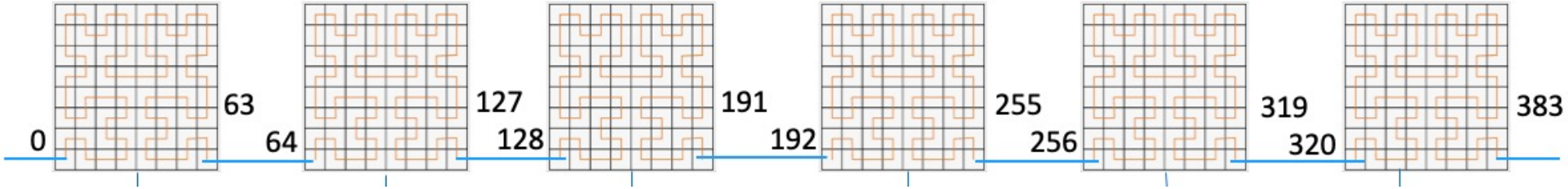


Processes have roughly same number of cells for comp/mem balance but must shuffle cells for AMR

- Time is explicit (a "file" for every time step) and that "file" contains all the state (for restart) (think 1 PB)
- Inside each mesh cell there is 10-100 state variables (64float) (temp, pressure, energy, momentum, …)
- 2D and often 3D – the other dimensions but how do you specify the geometric dimensions?



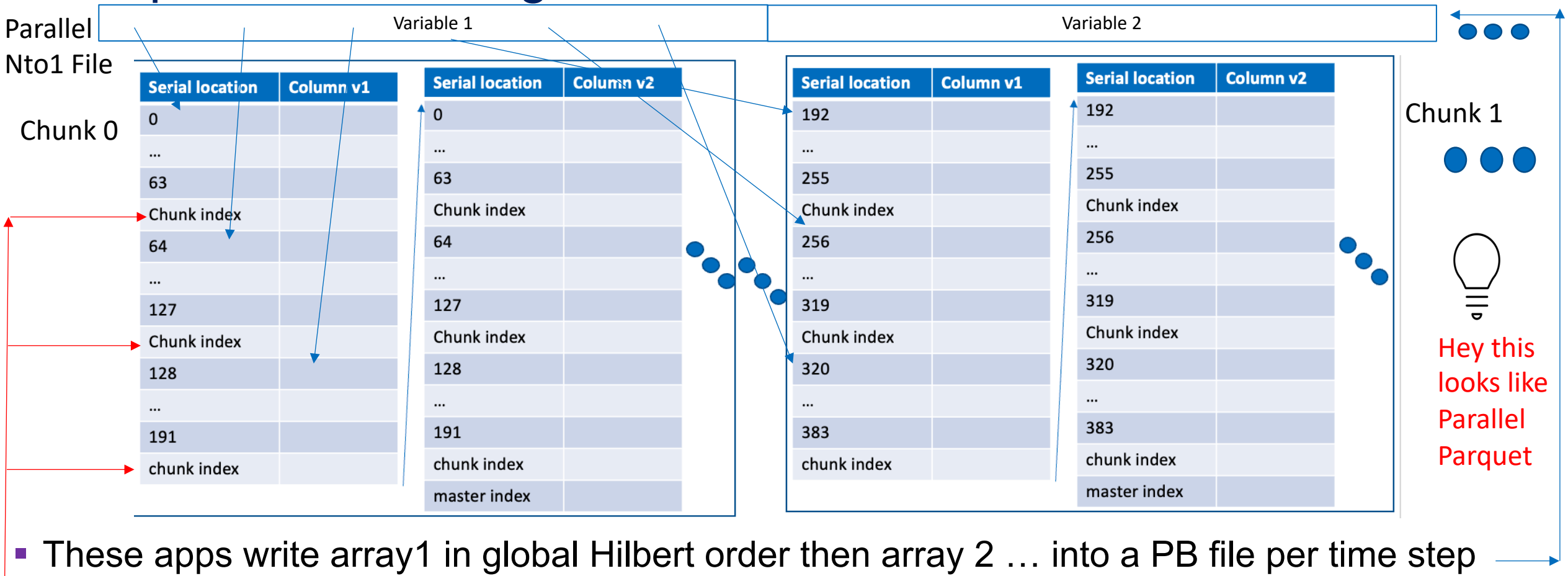
Single process Hilbert order



Hilbert space filling curves, we will call this the global Hilbert order, it serializes a value in the cells into distributed array. So we really have 10-100 distributed arrays in Hilbert order ☺

Find the outer edge of the eddy's (light blue and yellow) <1/100th of the total data, usually less. Can light weight indexing yield 1000X less data and can it be done very near the storage device to save transmission?

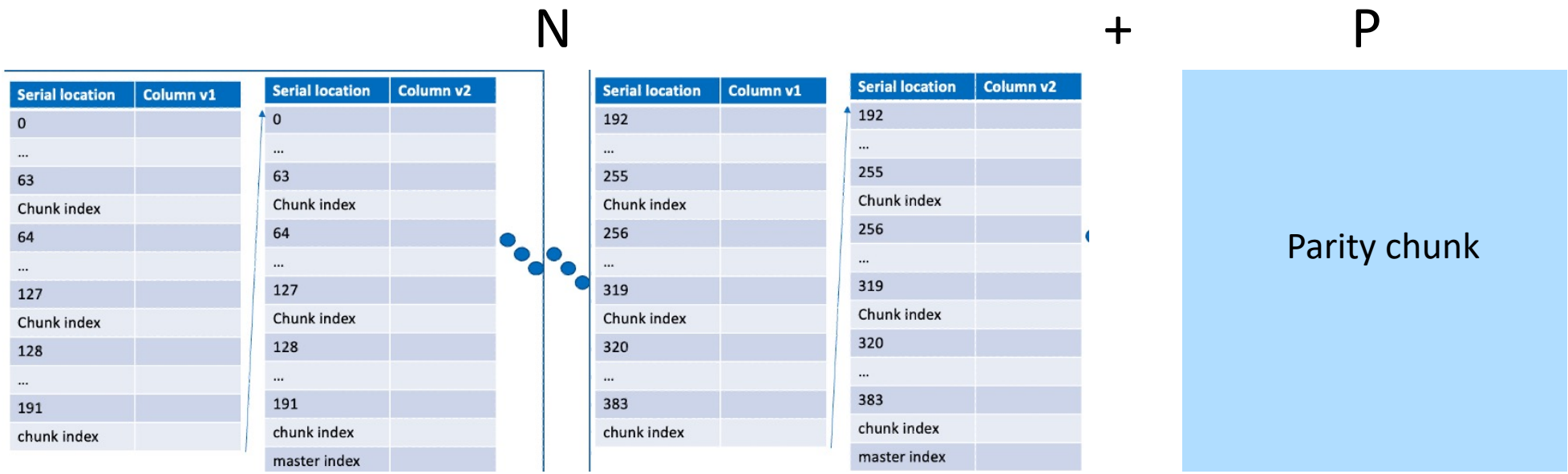
Adding index and offloading columnar analytics into Computational Storage, how would it work?



- These apps write array1 in global Hilbert order then array 2 … into a PB file per time step
- Adding light weight indexes for every chunk of every variable is doable
- Use standard analytics with things like Duckdb or Apache Drill and have the power of SQL and joins on columns and the simple indexes do massive reduction in parallel

Can this Hilbert Inspired Chunked Parquet Concept Extend to On-Disk Processing, Even with Erasure?

- Parquet ZFS File with Erasure and On-Kinetic Disk Analytics in parallel



A collaboration with our excellent partners at Seagate

- Arrange for your chunks to be the size of a parity stripe
- Use standard analytics with things like Duckdb or Apache Drill and have the power of SQL and joins on columns and the simple indexes do massive reduction in parallel

Related talks at FMS, SDC, SC22 and Food for Thought

- Related talks
  - FMS Ordered KV-CSD Prototype, SK hynix (Woo Suk Chung)
  - FMS Ordered KV-CSD Prototype Usage, LANL (Qing Zheng)
  - FMS ABOF 1.0, Eideticom, Nvidia, LANL (Dominic Manno)
  - SDC ABOF File System Offloads, Eideticom, Nvidia, LANL (Dominic Manno)
  - SDC ZFS Offloads to Computational Storage, LANL (Jason lee)
  - SDC Seagate Kinetic Scientific Columnar Query on ZFS Offload, LANL (Qing Zheng)
  - SC22 Grand Unified File Index ++, TBD
- Food for Thought
  - As we look into making NVMEOF a first class player in the HPC networking world we are:
    - Working with Fungible regarding scalability of NVMEOF endpoint state management and DPU REGEX use
    - Put SPDK on top of MPI – Why? - to enable massive scalability and performance testing over massive RDMA net
    - Looking for light weight security solutions for user space access to NVMEOF target/zone with no kernels involved
    - Beginning to look at routing NVMEOF RDMA between Ethernet, Infiniband, Slingshot, etc.

Caveat: Everything in this talk was harder to do than I made it sound ☺

Thanks for your time!

Ultra-Scale Systems Research Center

Los Alamos NATIONAL LABORATORY EST. 1943

NNSA

HIGH PERFORMANCE COMPUTING

EMC3

The Efficient Mission Centric Computing Consortium

Please take a moment to rate this session.

Your feedback is important to us.

