SNIA DEVELOPER CONFERENCE

SDC 24

BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

Storage for AI 101

A Primer on AI Workloads and Their Storage Requirements

Curtis Ballard (HPE) and Craig Carlson (AMD)

Introduction to Storage for AI 101

Presented by the SNIA Technical Council AI Taskforce

How SNIA can serve you with vendor neutral solutions to storage for AI gaps?

Speakers

Curtis Ballard

Hewlett Packard Enterprise

Craig Carlson

AMD

Bios available at: [SNIA 2023/2024 Technical Council]

SDC 24

What this presentation IS

A high level introduction to how AI uses storage

Foundational material for other AI presentations

Food for thinking about how to create storage for AI

What this presentation is NOT

Not education on AI techniques or Architecture

Not education on Storage techniques or Architecture

Not a deep dive into variations in storage use in AI

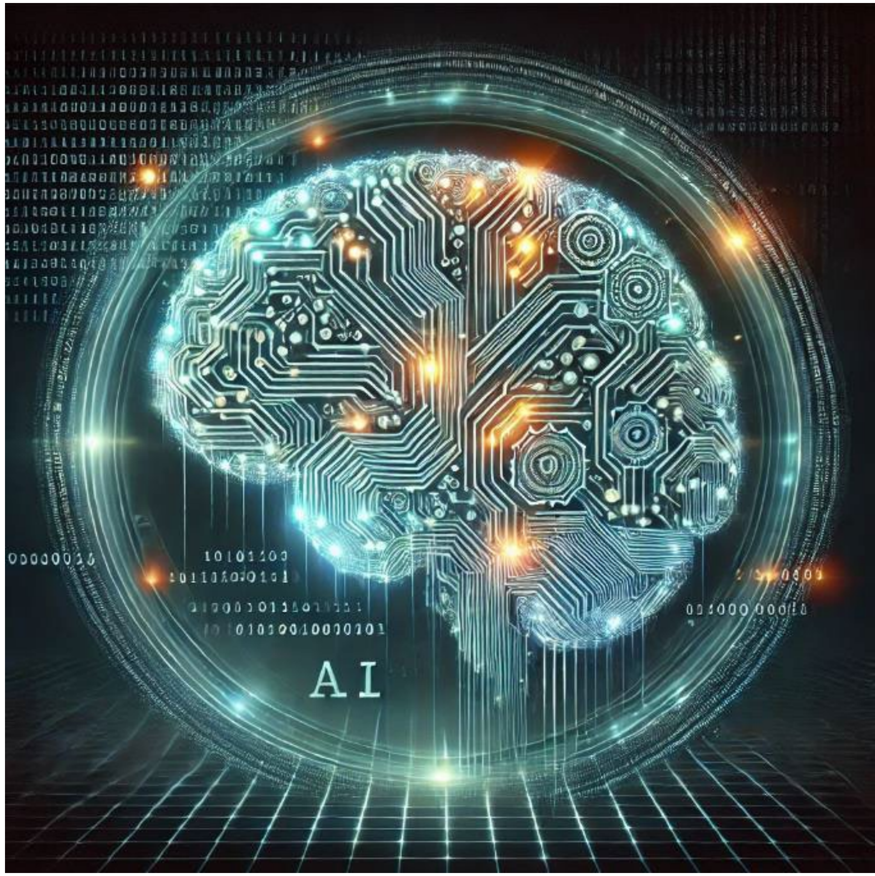
What Kind of Storage?

- Direct attached storage?
- External storage?
- Block storage?
- File storage?
- Object storage?

- Yes – all of the above
  - Storage needs have to be tuned, and will change, for different AI workloads and different model sizes

Terminology

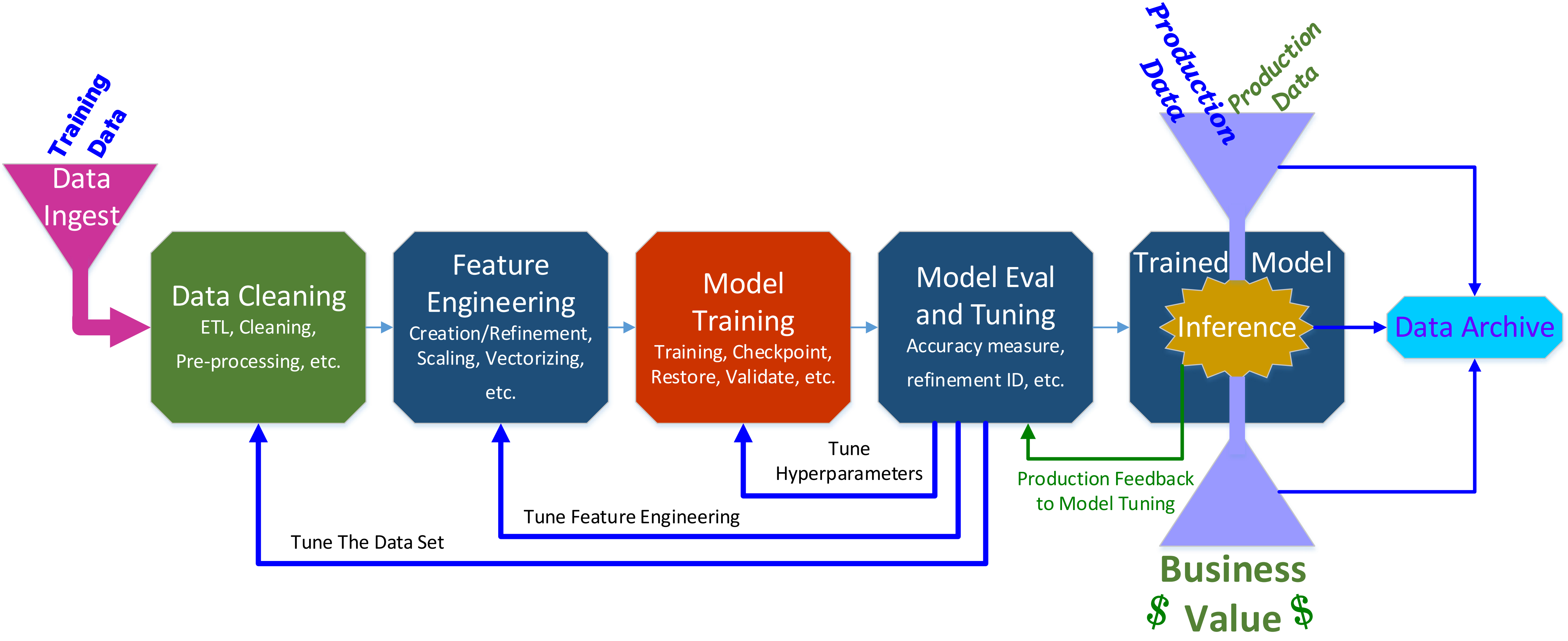
AI	A technique for leaking company secrets
Checkpoint	Storing the state during model training
Evaluation	Comparing model results to expectations
GPU	A tool to help secrets leak more quickly while consuming power
Feature Engineering	Transforming raw data into features for AI/ML models
Inference	Using a trained AI/ML model to analyze data
Large Language Model (LLM)	An AI/ML model that can generate and understand natural language
Retrieval Augmented Generation (RAG)	Using an external data source with a trained large language model to incorporate external data into generated output
Training	Teaching an AI/ML model to produce expected results from the input data
Tuning	Adjusting input to training based on evaluation of model output
Vector DB	A database optimized for storing lists of numbers (vectors)



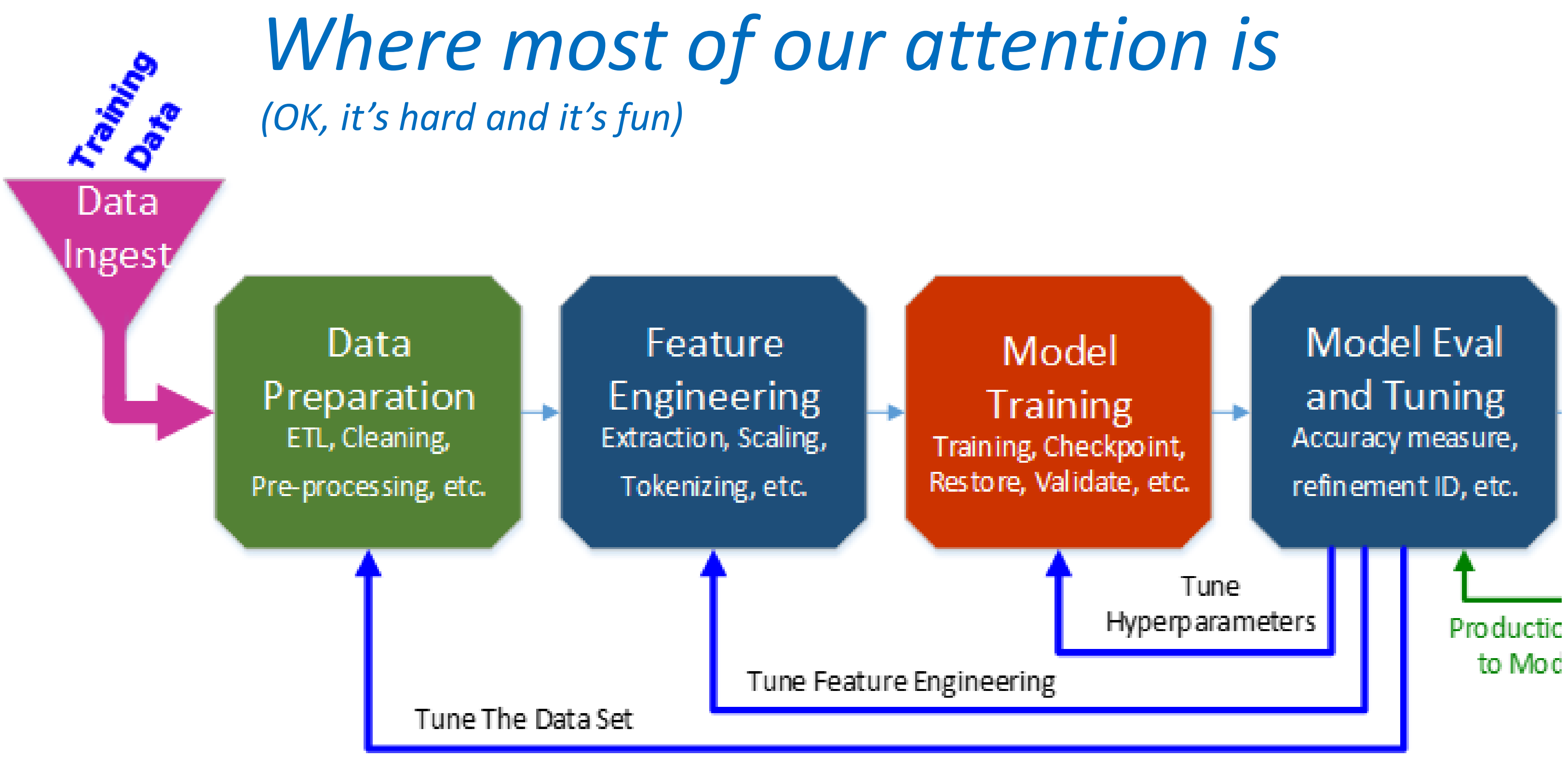
Why is storage for AI different?

- AI is a multi-phase workload
  - Most traditional workloads like databases have predictable access patterns
  - AI has widely different workload patterns for different phases
- Optimization goals may be different
  - Optimizing for GPU utilization instead of transaction response time
  - Optimizing for Data Scientist efficiency
- Highly parallel operations
- Performance and capacity varies widely for different AI tasks

Storage Phases of AI one perspective



Model building phases



Extensive use of:

- Data Scientists
- Compute Resources
- Storage Resources
- GPU Resources

With a goal of:

- Generating a Trained Model

Not generating business value unless your business is selling foundational models (e.g., LLMs)

Using AI to Generate Business Value

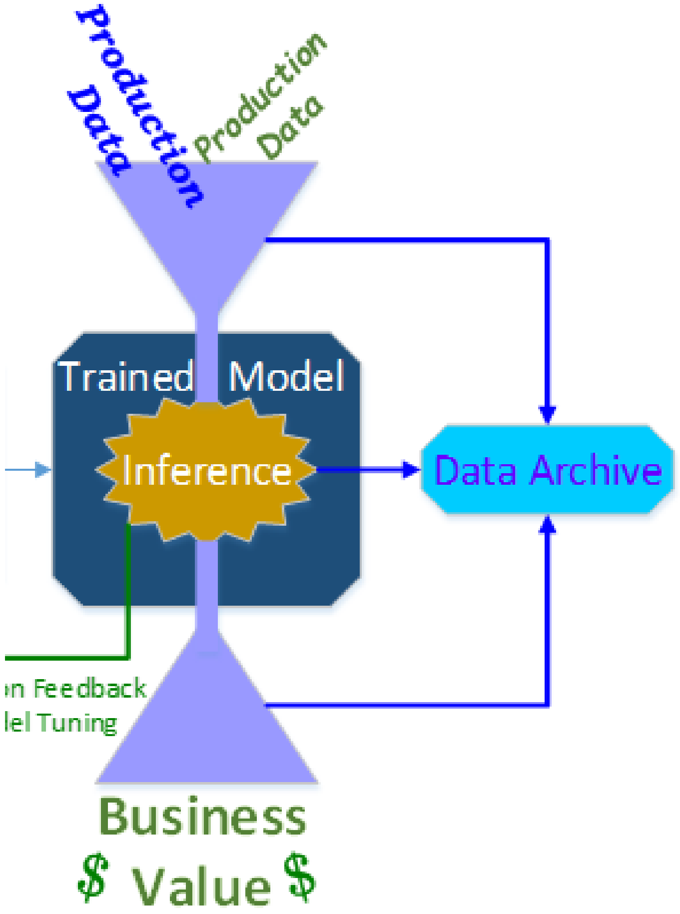
Where business value is generated

(it's still hard to get right and can still be fun)

Extensive use of:
- Production Data

More efficient use of:
- Compute Resources
- GPU Resources

With a goal of:
- Generating business value



How Does AI Change My Storage Needs?

Think about your needs for today and tomorrow

How does using AI change your storage requirements?

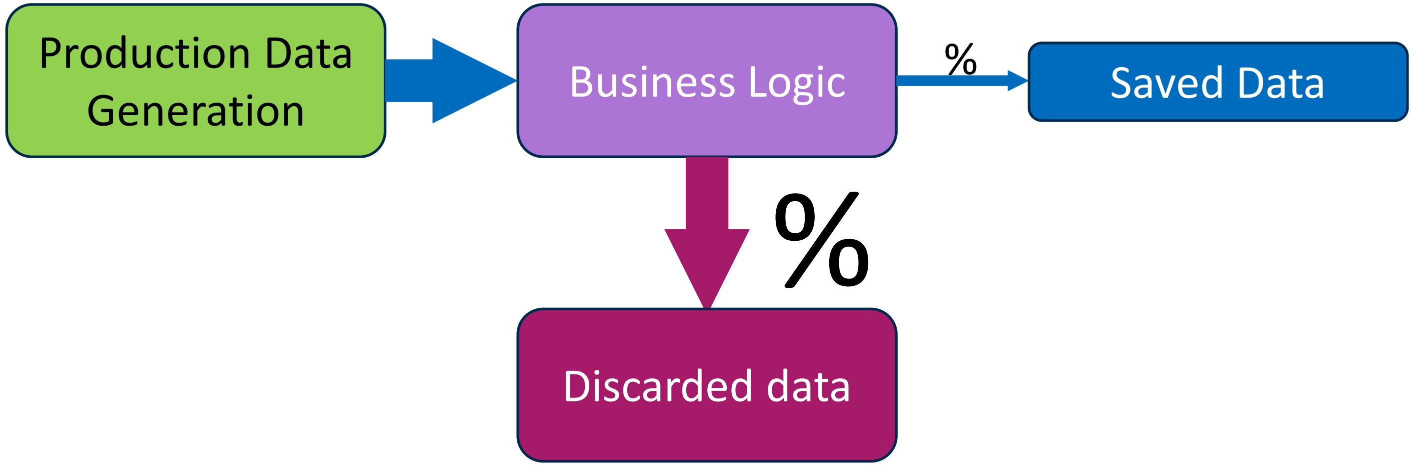
Example: Data Ingest

- Your business processes generate data today
- You already have storage for the ingest data
  - *or do you?*

- Business data is already being captured, ***But:***
  - How does AI affect what you capture
  - How does AI affect how you store your business data
  - How does AI affect how you access your business data

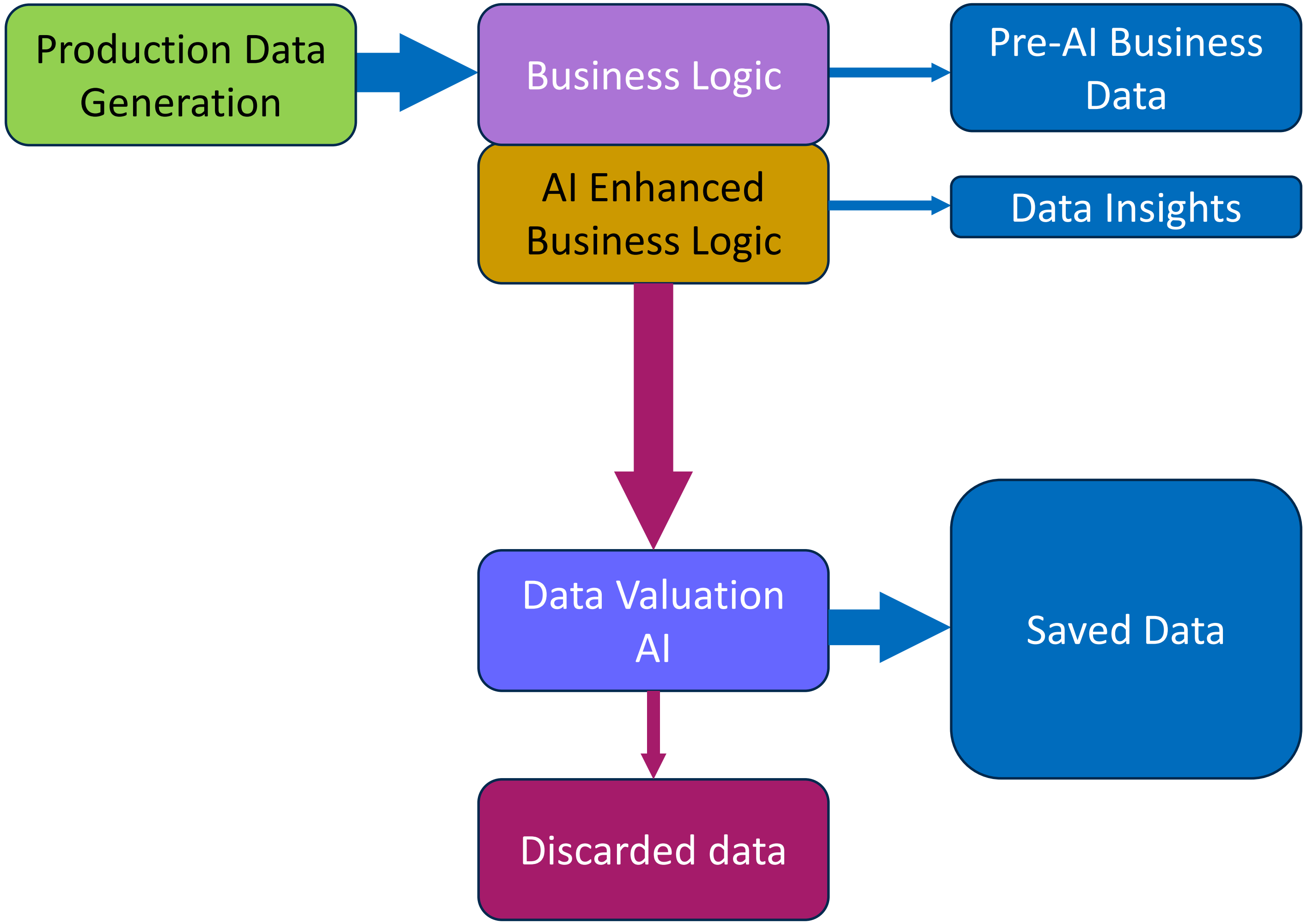
One Example Real Company

What their data ingest WAS before using AI



- Input data was mostly sequential write
  - rate determined by business data generation rate
- Random reads of small % of data
- Large amounts of data often discarded

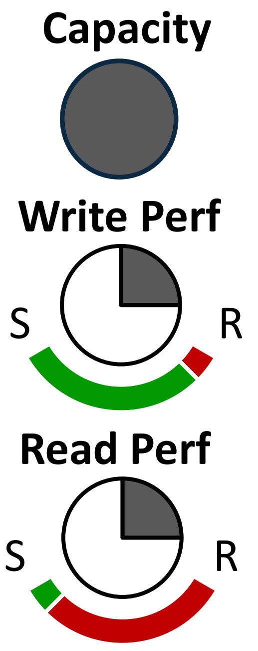
AI Uncovered Value in Data Previous Discarded



- Input data mostly sequential write
  - rate determined by business data generation rate
- Random reads *of* small % of data
- Random reads *from* large % of data
- Large amounts of data saved for future insights with improved AI Enhanced Business Logic

So what do the storage requirements look like for these Storage for AI phases?

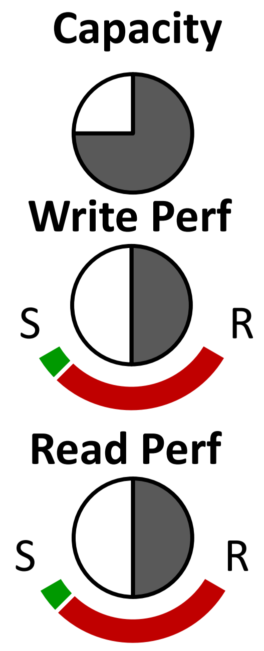
Data Cleaning



- Raw source data has to be prepared for use in AI
  - Logs, pictures, video, documents, etc.
- Data needs organized before becoming training data
  - Clean out noise
  - De-duplicate
  - Normalize
  - Privacy and Ethical processing, (anonymizing PII, removing bias, etc)
- Data is read from the ingest storage
- Cleaned data needs written to storage for Feature Engineering
- Process may be able to be partially automated applying AI

Sequential Random

Feature Engineering

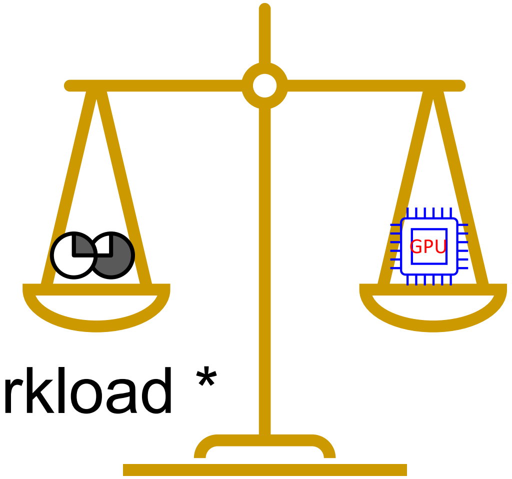


- Data Scientists serve as translators
  - Raw data → Food for AI (Numbers)
- Exploring the data – identifying patterns, outliers, relationships, etc.
- Splitting data for training and testing
- Feature extraction – converting key features into consumable nuggets
- Data transformation – converting data types (Vectorizing)
- Often highly parallel

Sequential Random

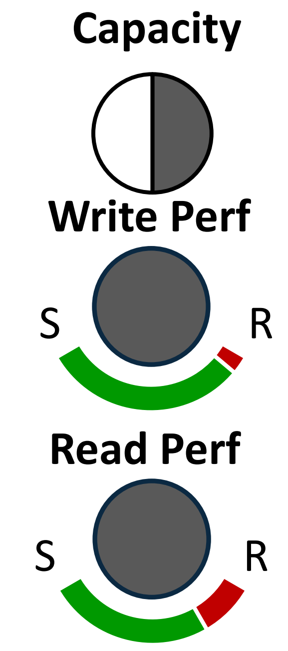
Model Training – General Storage Planning

- GPUs drive the cost – maximizing GPU utilization optimizes investment
- Design for a balanced architecture
  - Balance storage performance with GPU requirements
- Consider data sources
  - May require both file and object access
- If known training workloads – match storage performance to workload *
  - AI GPU benchmarks can show peak performance for various models
  - MLCommons MLPerf Training benchmarks is a good source
  - Determine size of training examples
  - Multiply throughput and size to estimate required read bandwidth
- For general purpose training may need to support GPU max read speed
  - Can be up to 1GB/s per GPU for high end GPUs today, increasing regularly



* https://www.snia.org/educational-library/storage-requirements-ai-2024

Model Training – When things can go wrong

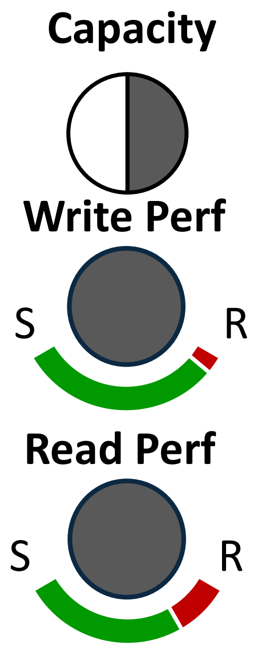


- Checkpointing – saving model weights and other state
  - Model weights are expensive when training takes a long time
  - Checkpointing saves state to allow restart after an error
- Checkpoint files are written sequentially
  - May be multiple sequential writes in parallel
- Training is paused – performance is money
- Checkpoint restoration is reversed
  - high sequential read, parallel reads to restore to multiple GPUs
- Storage performance determined be save/restore time goals

Sequential Random

Model Evaluation and Tuning

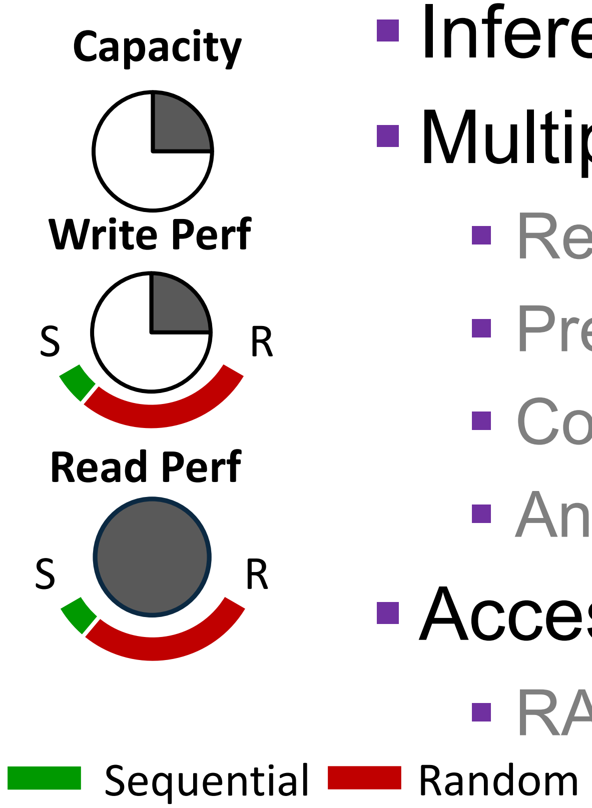
- Evaluation – measuring how well the results of the model match expectations
  - Accuracy – how often is it correct?
  - Precision/Recall – roughly a measure of how often wrong vs right
  - Measures such as F1 Score and AUC-ROC (area under the curve/receiver operating characteristics)
- Tuning – Adjusting hyperparameters to improve evaluation
- Produces a dataset containing the Model Parameters
  - Internal representation of the neural network
- Model Parameters size is constant, based on # of weights



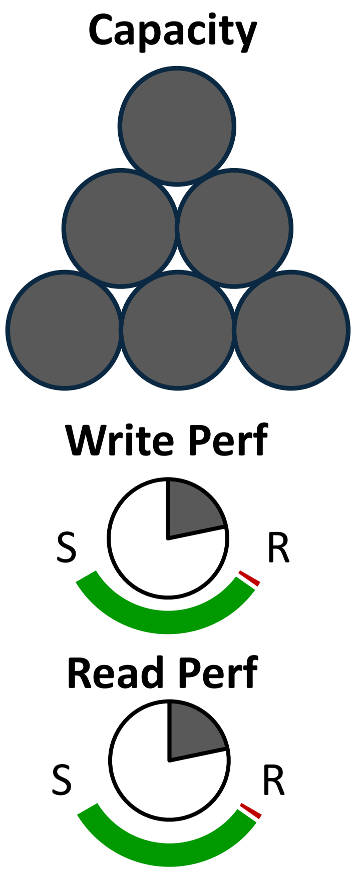
Sequential Random

Inference

- Running production data through the finished model to generate business value
- Inference = Inferring information from the data
- Multiple types of Inference
  - Retrieval Augmented Generation from LLMs
  - Predictive analytics
  - Computer Vision
  - Anomaly detection (e.g., malware, fraud)
- Access pattern can vary some depending on type of inference
  - RAG can produce a random workload similar to databases



Archive



- Often overlooked, not core AI but important for storage for AI
- Mandated by regulations for some AI applications
- Similar, but not traditional “archive”
  - Archived data may be brought back for training or new insights
- Performance needs vary but “just fast enough”
- No accepted terminology, maybe “Cold Storage”
- Continually growing data set
- Requires low cost and low carbon footprint storage
  - Opportunity for zero power storage such as DNA and Optical

Sequential Random

Tools and Technologies

Considerations for building your next infrastructure for AI

Calculating Performance

- Benchmarking
  - Publicly available AI benchmarks are available through ML Commons
  - Multiple categories
    - MLPerf Training
    - MLPerf Inference
      - Mobile
      - Tiny
      - Datacenter
      - Edge
    - MLPerf Storage
    - AlgoPerf: Training Algorithms Benchmark Results

Accelerators

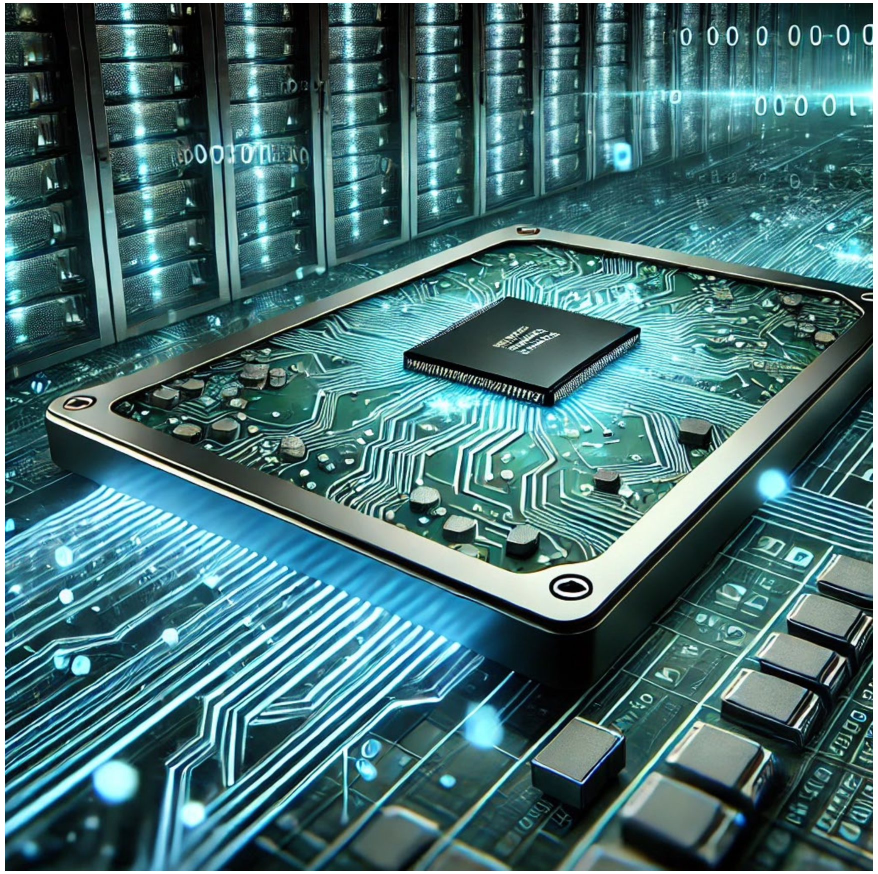
- SDXI
- Computational Storage
- GPUs

Accelerators – SDXI

- SDXI is a standard data mover being developed by SNIA
- Future versions of SDXI are looking to provide additional functions
  - Encryption/decryption
  - Compression/decompression

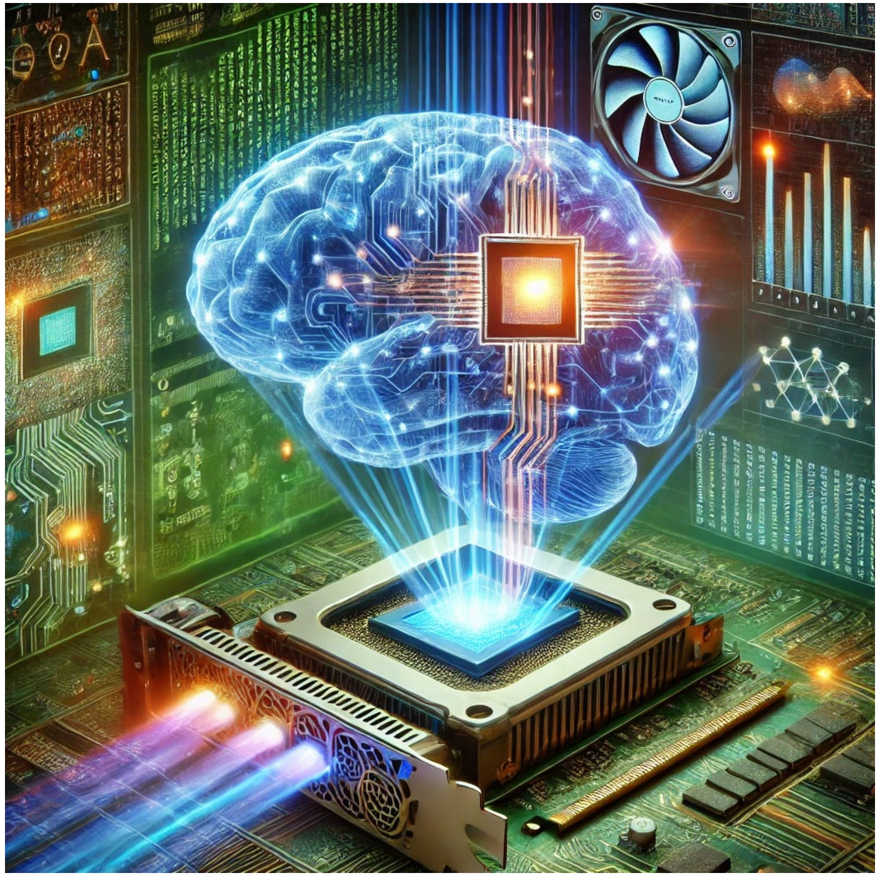
Accelerators – Computational Storage

- Computation Storage defined by both SNIA and NVMe
  - Open platform for adding computational functions to storage devices
  - Moves the computation closer to the data
  - Typical functions could be
    - Encryption/decryption
    - Compression/decompression
    - Data filtering
    - Data preparation for training



Accelerators - GPUs

- Parallel operations
  - AI calculations can be made highly parallel
    - Typically they are multiple similar calculations across a matrix
    - This is the type of calculation that GPUs are designed to handle in a massively parallel fashion
    - CPUs typically can only do a single calculation at a time
  - Not only do parallel operations reduce the computation time dramatically, but they also make it more energy efficient
- HBM - High speed memory typically found on datacenter GPUs



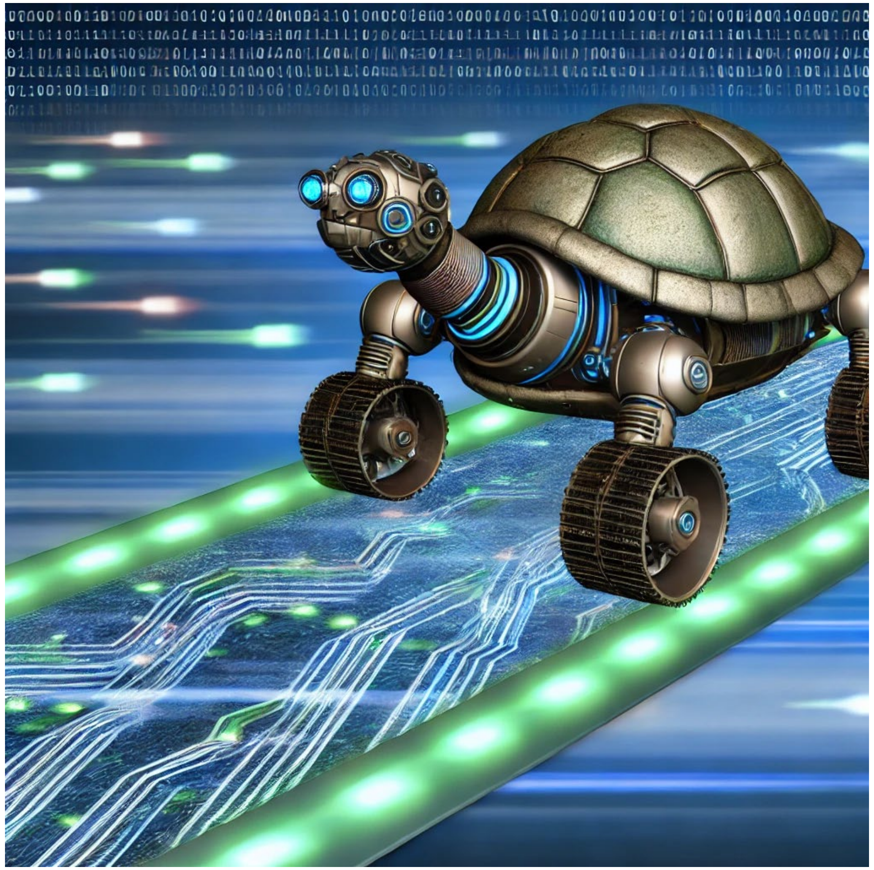
SDC 24

Accelerators - GPUs

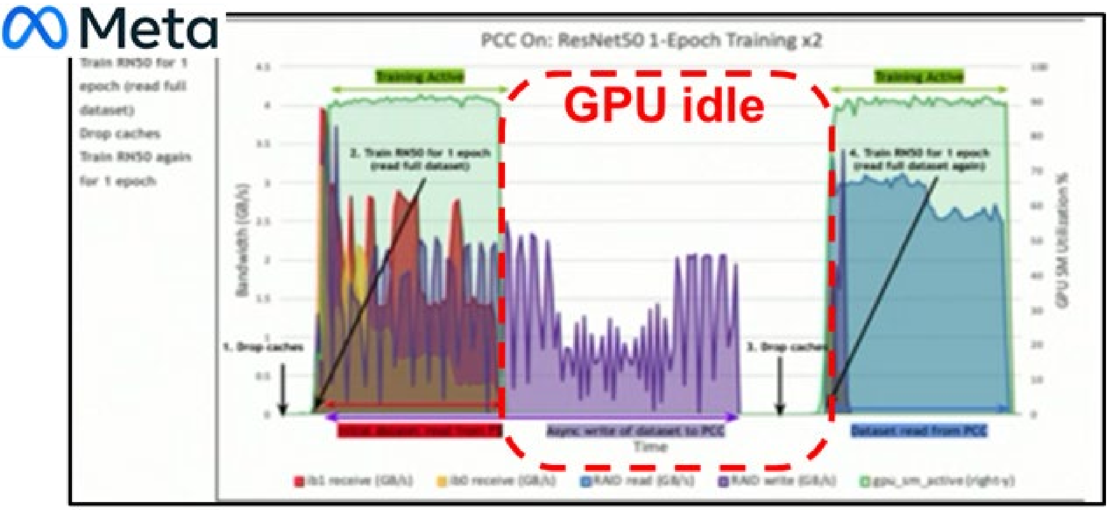
- GPUs are typically more effective for training, but they can be used for inference as well
- Downsides of GPUs
  - Harder to program
  - Can use a lot of power
    - Higher costs and cooling requirements
  - Expensive
  - Moving Data into and out of the GPU can introduce latency

Network (and Storage) Patterns

- Remember, you are only as fast as your slowest part
  - Due to inherent latency and device constraints… Network and storage components are often the slowest components in a system
  - Storage devices typically have slower access times
  - Networks are limited by latency
- The goal… Keep the GPUs fed!



Checkpoint Example



Meta's @scale Jun'24 Credit: NVidia

- Very typical for GPUs to wait on storage while checkpointing
- The GPU is typically the most expensive component in the system, so this isn't ideal!

Checkpoint Data Pattern

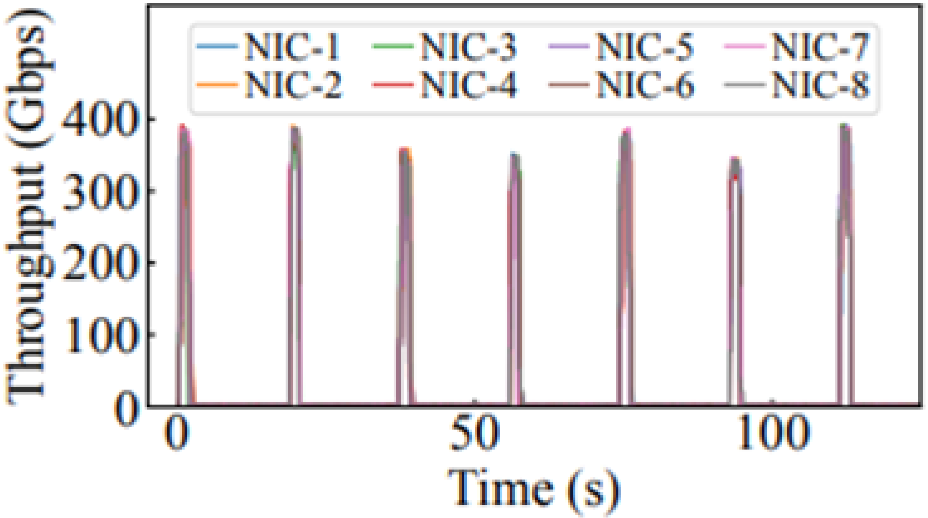
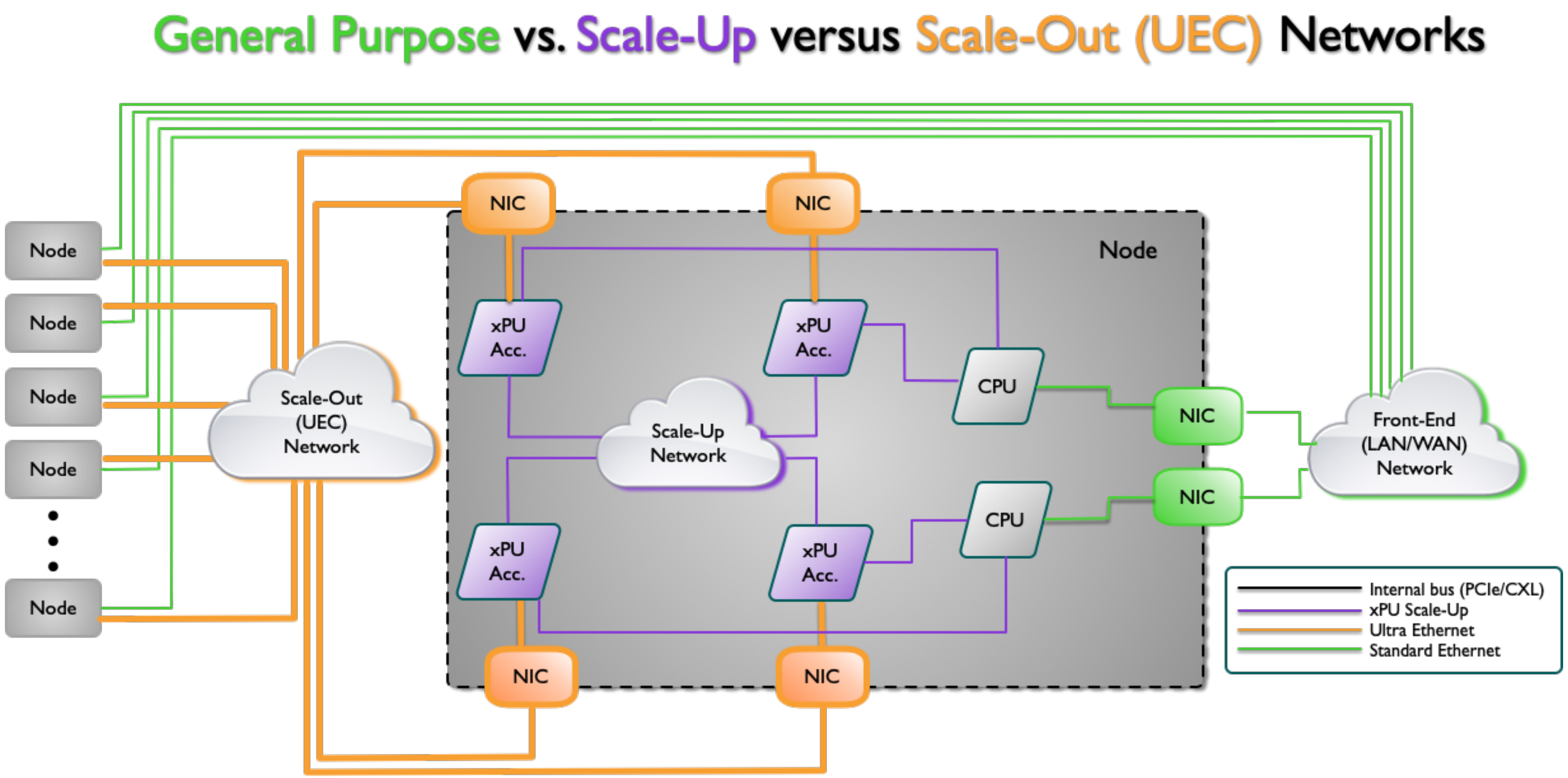


Figure 2: NIC egress traffic pattern during production model training.

- This results in very bursty network data patterns (for network based storage)

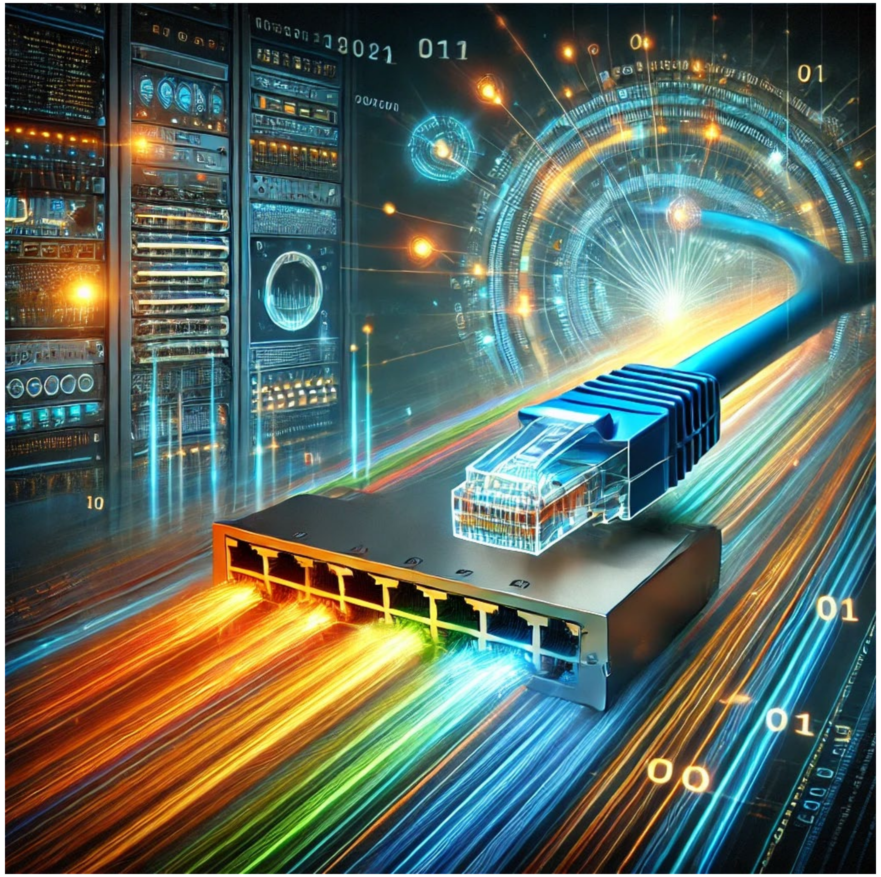
Which Network?



- Scale-out network is the “backend” to the GPUs for communications between racks
- Many proprietary solutions exist
- Ultra Ethernet (UEC) is an open solution under development

Scale Out - Ultra Ethernet

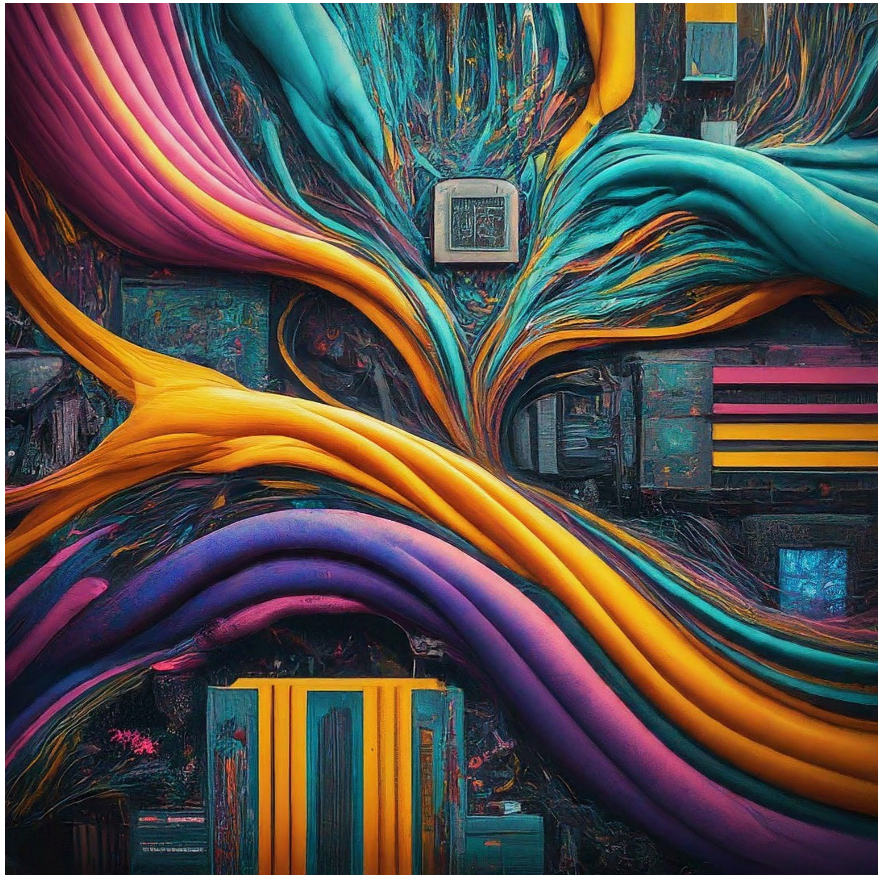
- Open project being developed under the Linux Foundation
- Goals
  - Highly scalable – Scale to a million nodes
  - The most recent congestion management techniques built-in
  - Low latency protocol design at both on the link layer and transport layer
  - Highly reliable with built-in error recovery
  - Security designed in from the beginning (not an afterthought)
- Ultra Ethernet taps in the expertise and experience of multiple members to develop and use the most recent technologies
- Specifications will become publicly available near the end of the year



SDC 24

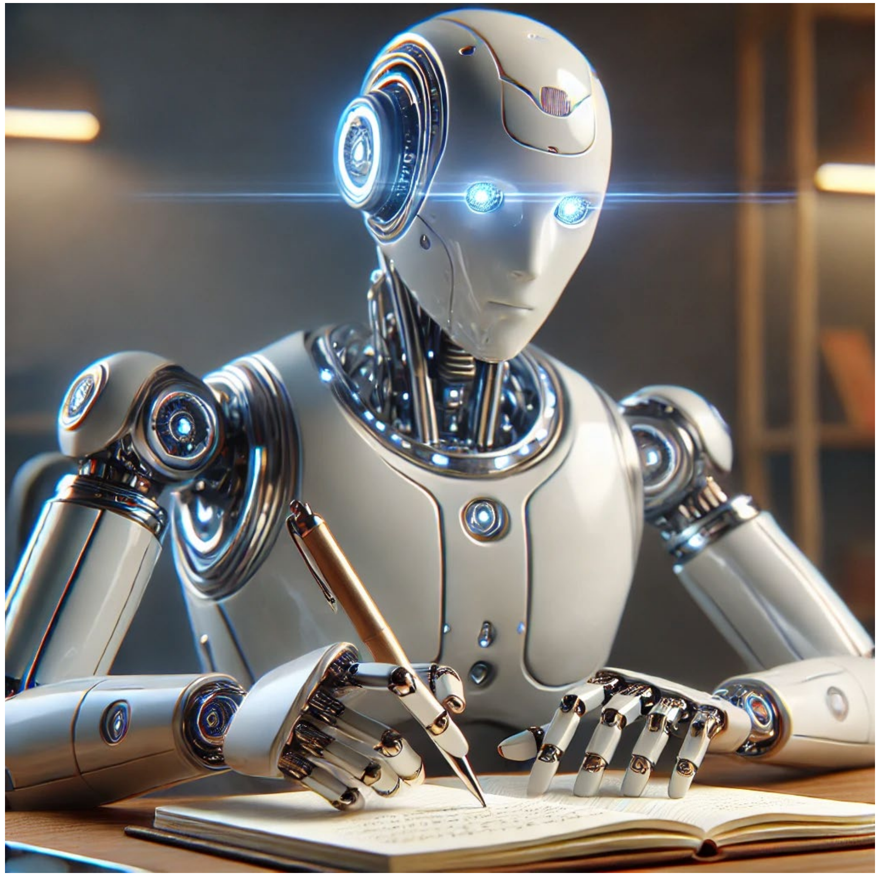
Scale Up – Ultra Accelerator Link

- The UALink interconnect is for scale-up Accelerator-to-Accelerator communication
  - The initial focus will be sharing DDR & HBM memory among accelerators
- Direct load, store, and atomic operations between accelerators (i.e. GPUs)
  - Low latency, high bandwidth fabric for 100's of accelerators in a pod
  - Simple load/store semantics with software coherency
- Supports data rates up to state-of-the-art 200Gbps per lane
- The initial UALink spec taps into the experience of the Promoters developing and deploying a broad range of accelerators and leverages the proven Infinity Fabric protocol
- Complementary with scale-out approaches such as Ultra Ethernet Consortium (UEC)



Storing it all

- Three types of storage typically used for AI
  - Cloud
  - Object
  - Block
- Model data (input and output) typically stored in the cloud or on object storage
- Block storage often (but not always) used for checkpointing
  - Low latency/high performance

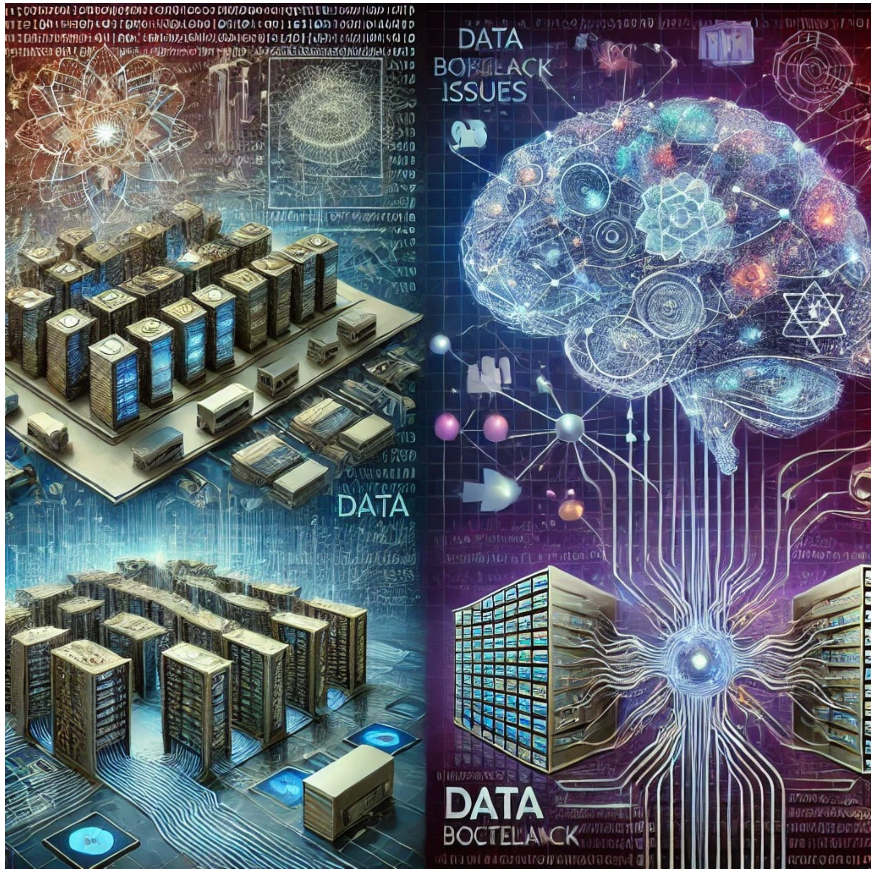


Enter CXL

- Additional Storage Functions could be provided by CXL attached memory pools
  - Allows a tiered memory where some, non-immediate use data, could be stored in a CXL pool
- CXL and other new memory architectures could provide relief to the existing memory bottleneck (or the “memory wall”)

Storage challenges

- Performance
  - Storage should take as little time away from the GPUs as possible
- Scalability
  - Model sizes continue to grow (almost exponentially!)
- Reliability
  - Does no good if checkpoint data is lost
- How could SNIA help with these?



SNIA Data pattern repository

- SNIA has an IO trace repository used extensively for research
  - The SNIA I/O Traces, Tools, and Analysis repository, IOTTA https://iotta.snia.org
- The repository does not yet have AI Storage workload traces
  - A gap SNIA would like to fill
- Please consider sharing any IO trace data you have with SNIA IOTTA so we can start building a repository for AI traces

Please take a moment to rate this session.

Your feedback is important to us.

