Using MRAM to Create Intelligent SSDs

Jérôme Gaysse

Senior Technology&Market Analyst

jerome.gaysse@silinnov-consulting.com

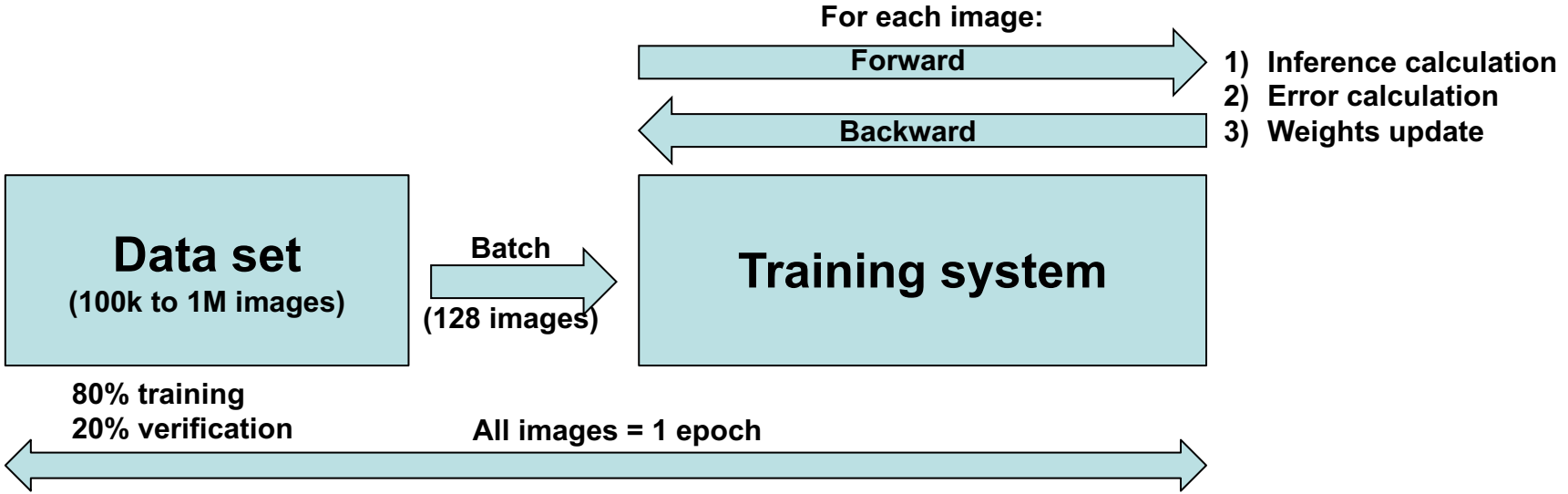
Study context

- Analysis of system & application
- Performance modeling
- Emerging technologies analysis
- New architecture definition
- Performance simulation

Deep learning

- Inference and training
  - Training value = weights value
- Training problem
  - Take a long time: time to market impact
  - Expansive hardware resources : TCO impact
- How MRAM could solve it?

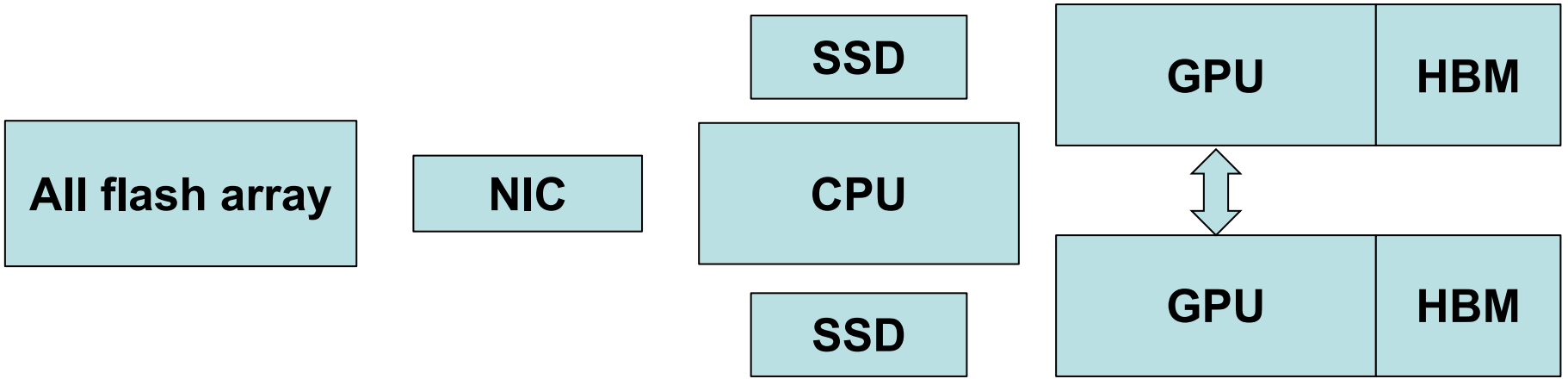
Training process



Training = multiple epochs

Deep learning training

- System architecture



Performance analysis

- Focus on ResNet50 neural network
  - 50 layers
  - 25M parameters
  - 3.9GMACs
- Many benchmarks available
  - Nvidia, HPE, Dell…

Performance analysis

- Throughput : 400 images per second/GPU
  - FP32 resolution
  - 25M parameters => 100MB model size
- Checkpointing (about every 400 images)
  - =>100MB/s write
- Data set : 100kB images @400fps
  - => 40MB/s read

No IO storage bottleneck

DL improvements

- From FP32 to FP16 to INT8
  - Less computing requirements
  - Less memory bandwidth requirements
- Pruning (less connexions between neurons)
  - Less computing requirements
  - Less memory bandwidth requirements

Training throughput to increase

Training performance increase

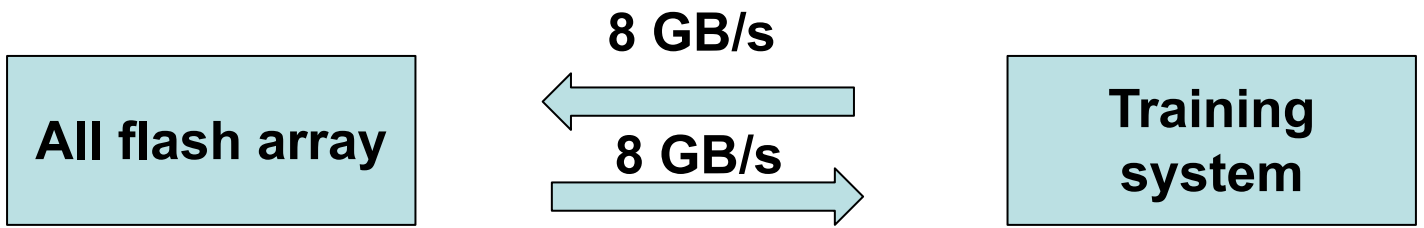
- With DL optimization,
- and new Deep Learning Processor developement

- From 400 FPS to 10,000FPS (estimation)
  - x25

IO impact

- If throughtput x25
  - Read => 1GB/s (x25)
  - Write => 1GB/s (x10)
    - x25 accesses but less data to write
- Average bandwidth, not so high
  - The key bottleneck will be the latency

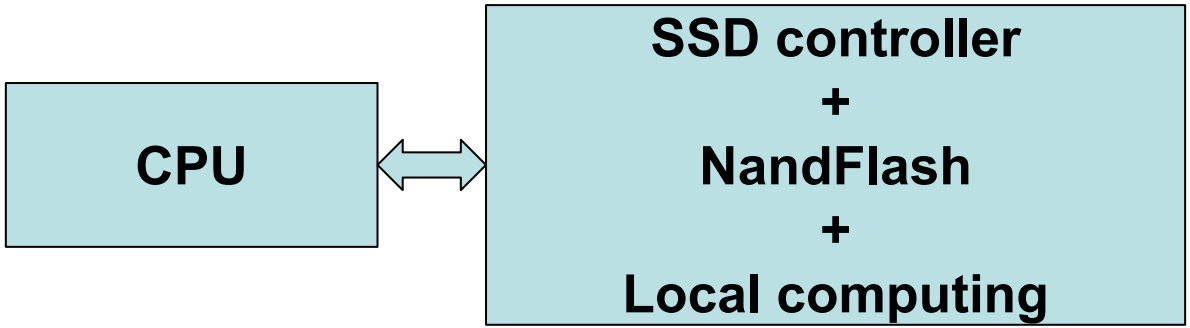
Need new architecture



- Huge data movement between training system and storage array
  - => High power consumption
  - => High silicon cost (AFA controller, NIC…)

Computational storage concept

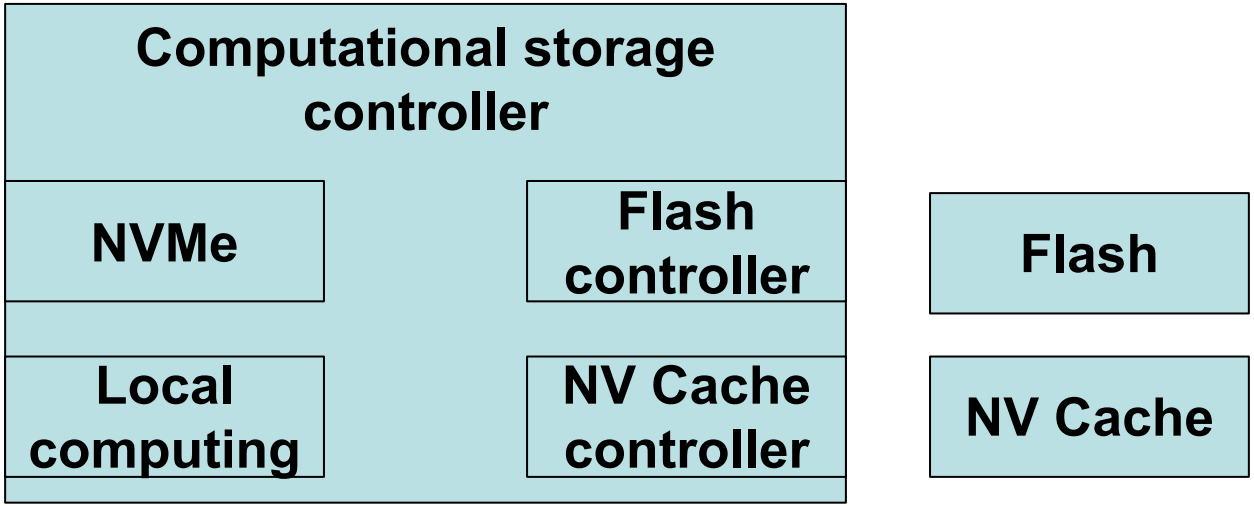
- Computational storage = computing capabilities in the SSD



Reduce data movement:
Less power, higher performance

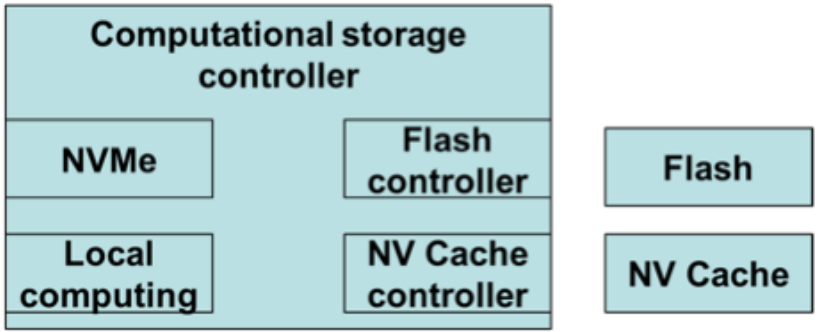
Computational storage architecture

- NV cache for DL weights checkpointing



Theory of operation

- Namespaces
  - Weights
  - Data set
- Vendor specific commands
  - Start training



NV cache technologies

- MRAM standalone chip

or

- NVDIMM-N like technology (DRAM+FLASH+Energy source)

Is MRAM capacity ok?

- Model size examples
  - 25MB to 100MB for RestNet50
  - 60MB To 230MB for RestNet152
- MRAM capacity
  - 256Mb chip today
  - 1Gb this year
  - 4 Gb in 2-5 years

Capacity: ok

Is MRAM performance ok?

- Weights Write : 1GB/s
- MRAM bandwidth:10GB/s
  - 1333MT/s

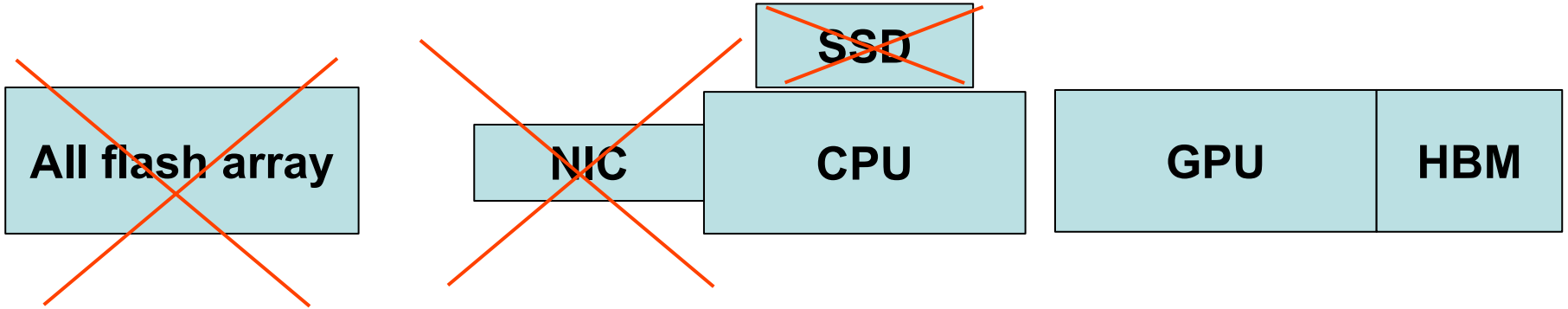
Performance: ok

Is MRAM endurance ok?

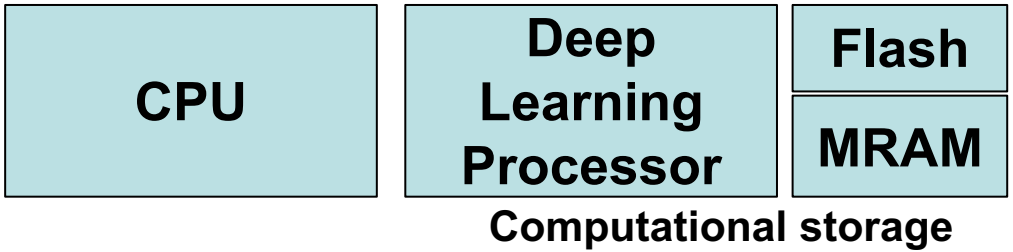
- Training setup
  - 50 epochs,
  - 1M images data set
  - Checkpoint every 400 images
  - =>125k write operations in 1.5h
- MRAM: $1x10^{10}$ cycles
  - 80k trainings, 13 years

Endurance: ok

System benefit

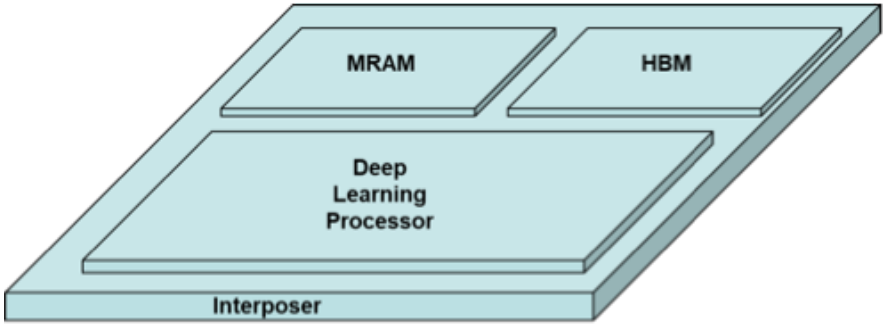


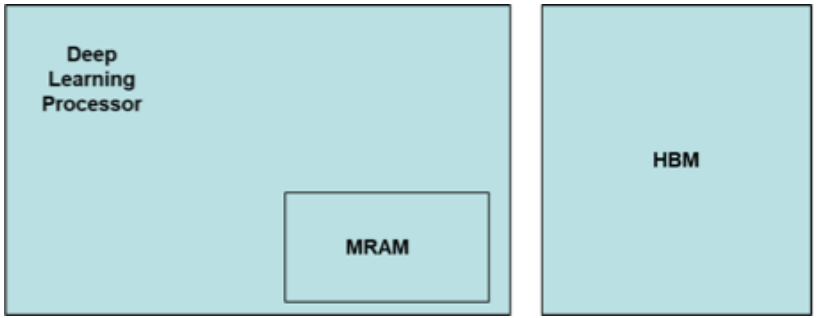
- Removing hardware cost & power



What about a more integrated solution?

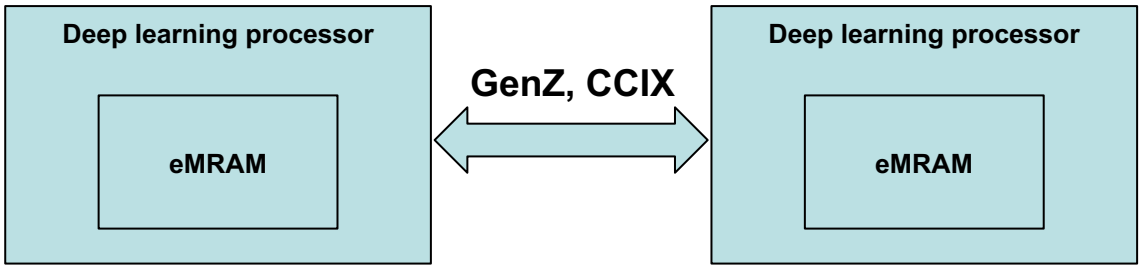
- 3D
- Embedded





Embedded NVM or Cache?

- Up to 32MB-64MB
  - Not enough to store all the weights
- New interconnect may solve it



Next studies

- Simulation at system level
  - Detailed storage access (latency)
- Computational storage applied to NVDIMM?

Want to know more?

Intelligent Hybrid Flash Management

Tuesday Aug 7, 3:40-5:45 PM, SSDS-102-1: Enterprise SSDs (SSDs Track)

A Comparison of In-storage Processing Architectures and Technologies

Monday Sept 24, 10:35 AM - 11:25 AM