Remote Persistent Memory - RPM
The Case for Use Cases

Paul Grun

Cray, Inc

Chair, OpenFabrics Alliance

Remote Persistent Memory

- Remote Persistent Memory is something different

- It might prove to be a transformative technology
- It's going to take some thought

Objective – Drive Adoption of Remote Persistent Memory

Something Different?

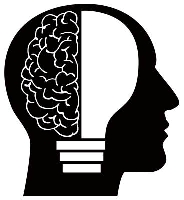
"Remote Persistent Memory is something different"

Different? Yes, because it involves a fabric

Ultimately, we need to talk about fabrics, and what is needed to make RPM useful

Hint: latencies in the network software stack are going to turn out to be very important

“It’s going to take some thought”



Starting by thinking about how RPM will be used

Hence, these four talks:

- Use cases for RPM
- RPM in the commercial space
- RPM in an HPC world
- What it might mean for the fabric

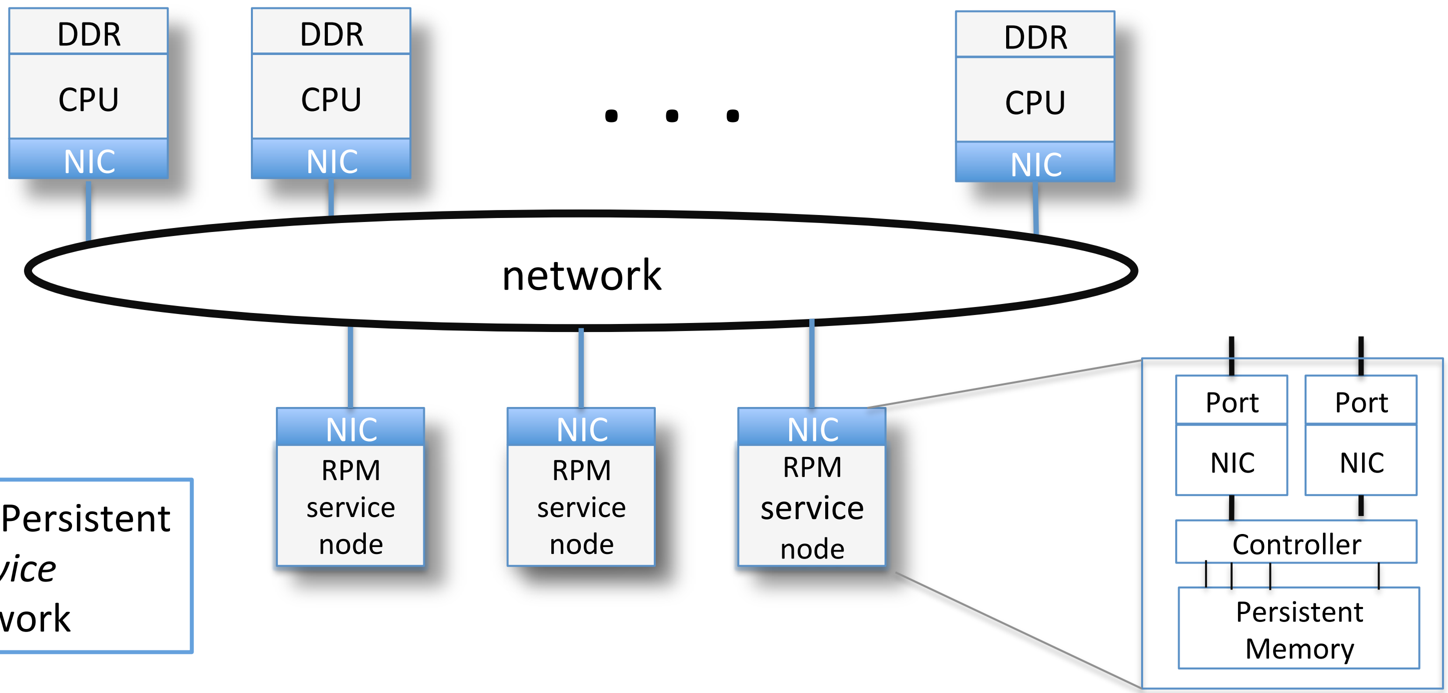
What is Remote Persistent Memory Exactly?

- **Locality**
  - A PM device accessed over a network
  - ~~A local PM device attached to an I/O bus or a memory channel~~

- **Access Method**
  - Persistent Memory as a target of memory operations (hence, 'memory')
  - ~~Persistent Memory as a target of I/O operations e.g. NVMe~~

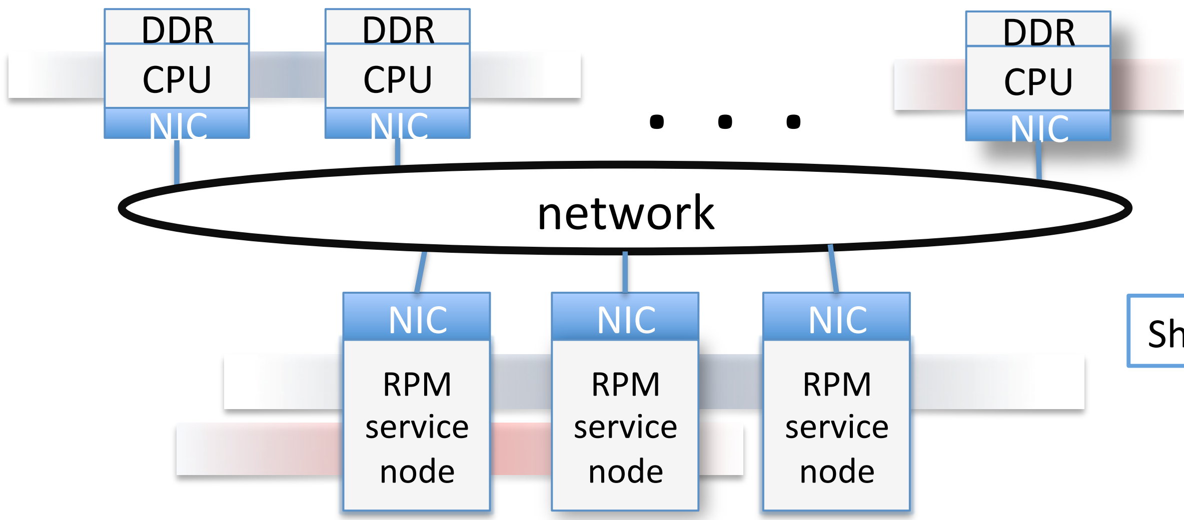
- **Memory Hierarchy**
  - Not as fast as local DRAM, but much faster than other remote technologies
  - Think of it as another layer in the memory hierarchy
  - Viability of RPM as a memory technology depends on how fast it can be accessed.

System Perspective



Think of Remote Persistent Memory as a *service* located on a network

Memory Model

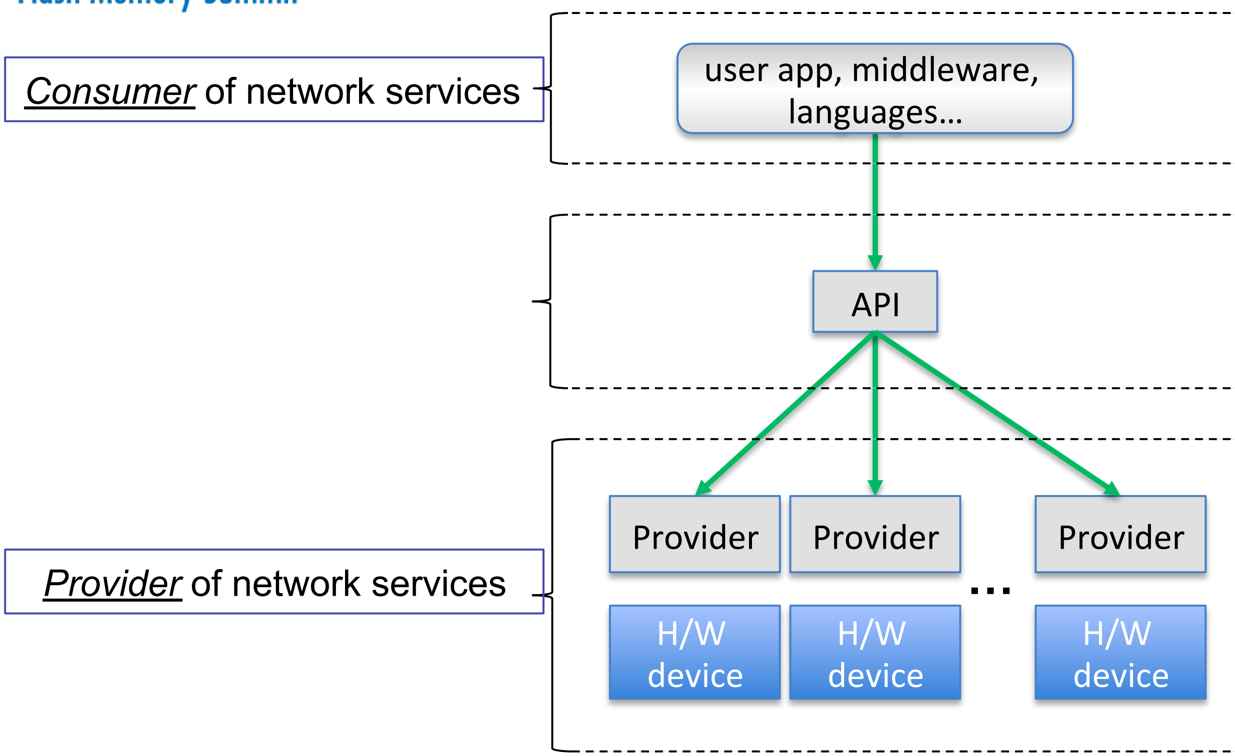


Shared or unshared resource

Organized into pools, accessed as memory

Can be configured as a flat address space, or as object storage. Or both.

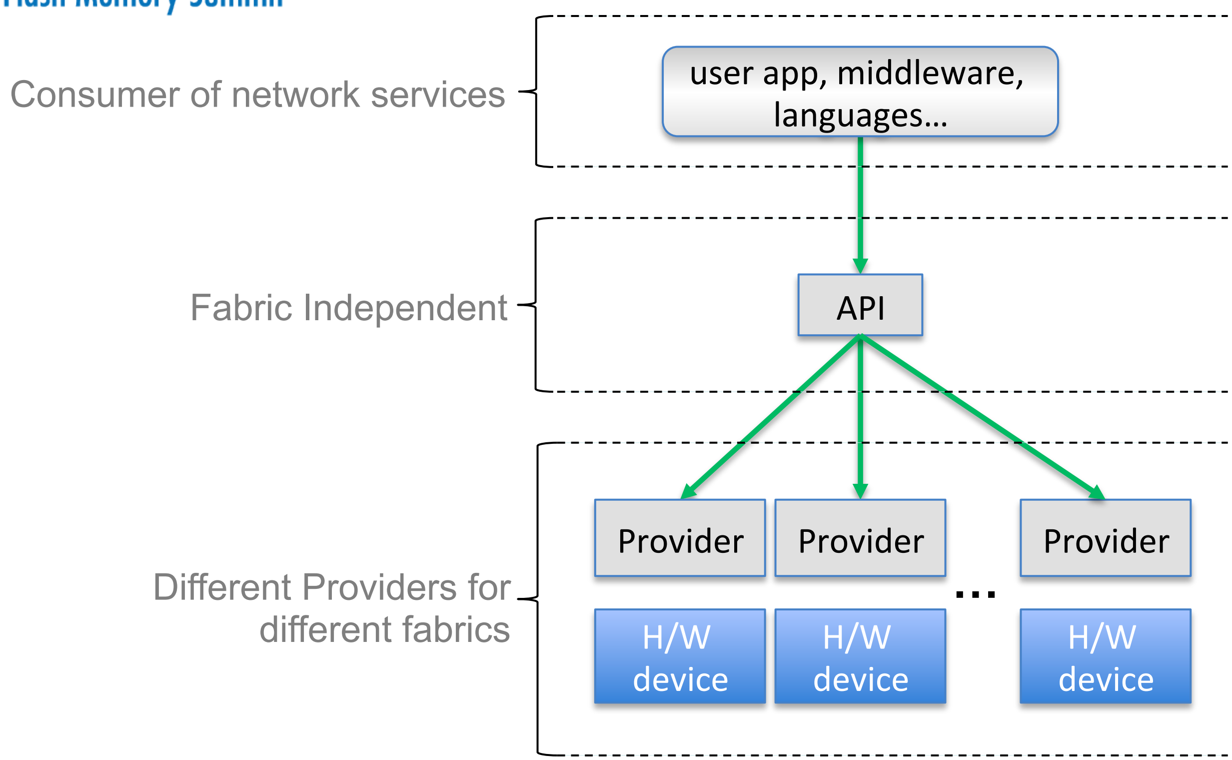
Some Taxonomy



To design the network, we're going to need to know something about the consumer

8

Top Down Design Begins with Use Cases



Let consumers drive the requirements

Define the functions to be exported upward

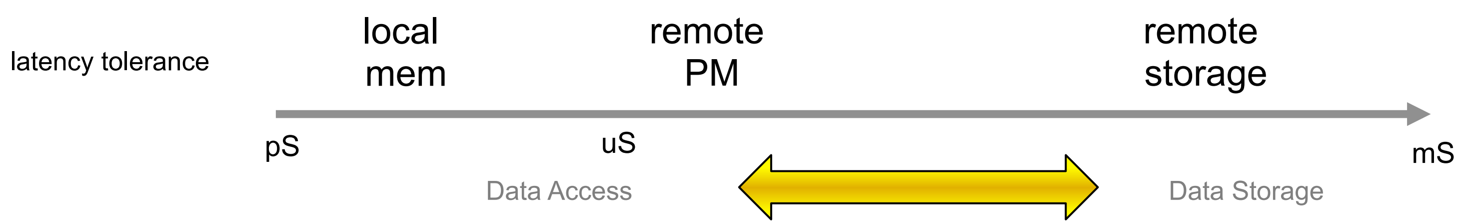
Implement APIs for the fabrics of interest

Why Focus on APIs?

- RPM will never be as fast as local memory
- Think of it as a new layer in the memory hierarchy
- We can't do anything about the speed of light…
- But we can reduce latency in the network stack

A Bold Prediction

Remote Persistent memory will change the way that applications store, access, communicate and share information



But only if the end-to-end latency can be kept very low

A Multi-dimensional Problem

To craft a network solution, and particularly to optimize the network software stack, there are number of factors to consider:

- Consumer considerations
  - For what purpose is the consumer storing/accessing persistent data remotely?
  - Under what conditions are data shared?
  - What is the security model?
- System objectives
  - For any given system, what are its design objectives? Performance? Scalability? High Availability?
  - What type of service is being offered? Object store? Pools of Memory?

Possible System Objectives

- High Availability
  - Replicate local cache to RPM to achieve high availability
- Scale out
  - Scale out distributed database or analytics applications
- Scale up
  - Scale up databases that exceed local memory capacity
- Disaggregation / independent scaling of memory and compute
  - Compute capacity scales independently of memory capacity

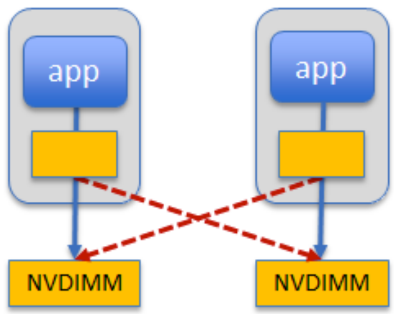
Some Consumer Considerations

- Application Objectives
  - Persistence vs capacity?
- Sharing Models
  - Shared data vs unshared data?
  - A shared service vs a dedicated service?
- Memory Model
  - Flat address space vs object stores?
- Characteristic Traffic Patterns, Traffic Engineering Requirements
  - Small byte operations vs bulk data transfer?
- Ordering Semantics, Atomicity

Possible Application Targets

- Scale up Databases
  - Operate on datasets larger than would fit into traditional memory
- Scale out Databases
  - Creating a common data store shared among database instances
- Graph Analytics
  - Operate on larger graphs than would fit in local memory
- Commercial Applications
  - Promote collaboration on large scale projects
- HPC Applications
  - Scalability, parallel applications

Example: High Availability

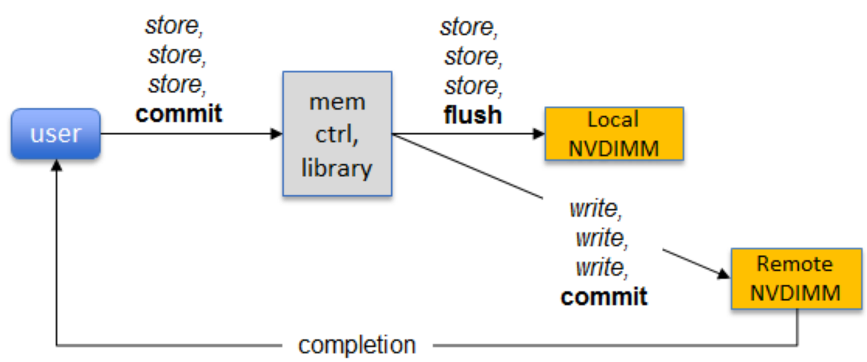


What it looks like

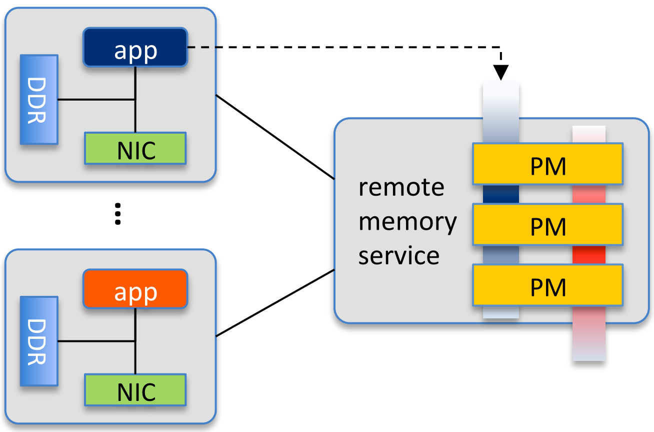
“High Availability”

Usage: replicate data that is stored in local PM across a fabric and store it in remote PM

How it works



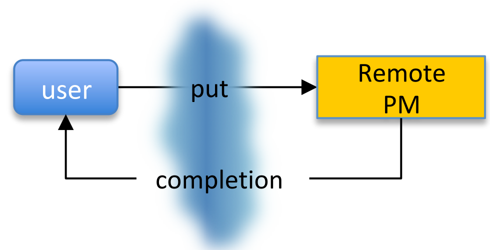
Example: Remote Persistent Memory



What it looks like

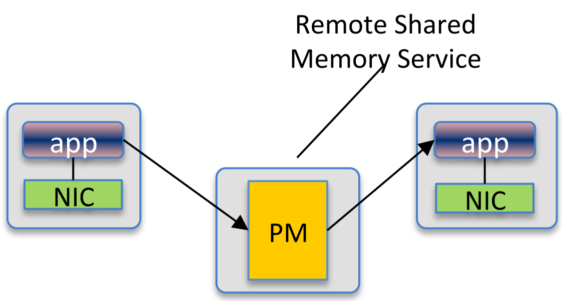
Usage: Expand on-node memory capacity, while taking advantage of persistence (or not). Disaggregate memory from compute.

How it works



"Scalable Memory"

Example: Shared Persistent Memory

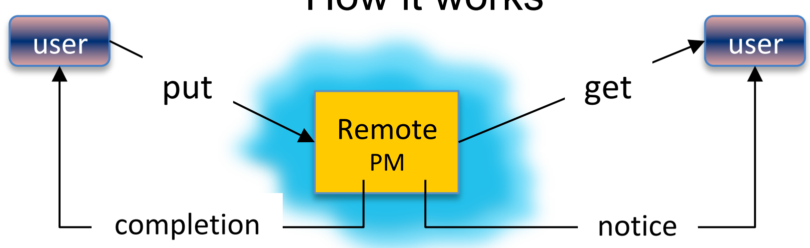


What it looks like

“Scale-out Applications”

Usage: Information is shared among the elements of a distributed application. Persistence can be used to guard against node failure.

How it works



An Example: RPM for Graph Analytics

- Operate on larger graphs than would fit in local memory
  - Solve Petabyte-sized graph problems on 1,000 nodes vs 10,000 nodes
- Persist data structures between program executions
  - Run multiple query jobs sequentially and potentially in parallel
- Use existing programming models and languages
- Make better use of available DRAM for algorithms, not just holding data
- Alternatives
  - Limit the size of graphs one can study to what fits in memory
  - Use out-of-core methods which store graph data structures on disk
  - Store graphs in large NoSQL database, write new algorithms

Collaboration

SNIA and the OpenFabrics Alliance are collaborating to drive adoption of RPM technology

Driving Adoption of RPM

<u>Programming Models</u>
- A common understanding among application developers of the behaviors that are required to reliably access Remote Persistent Memory,

SNIA

<u>APIs</u>
- The means for an application to implement those required behaviors

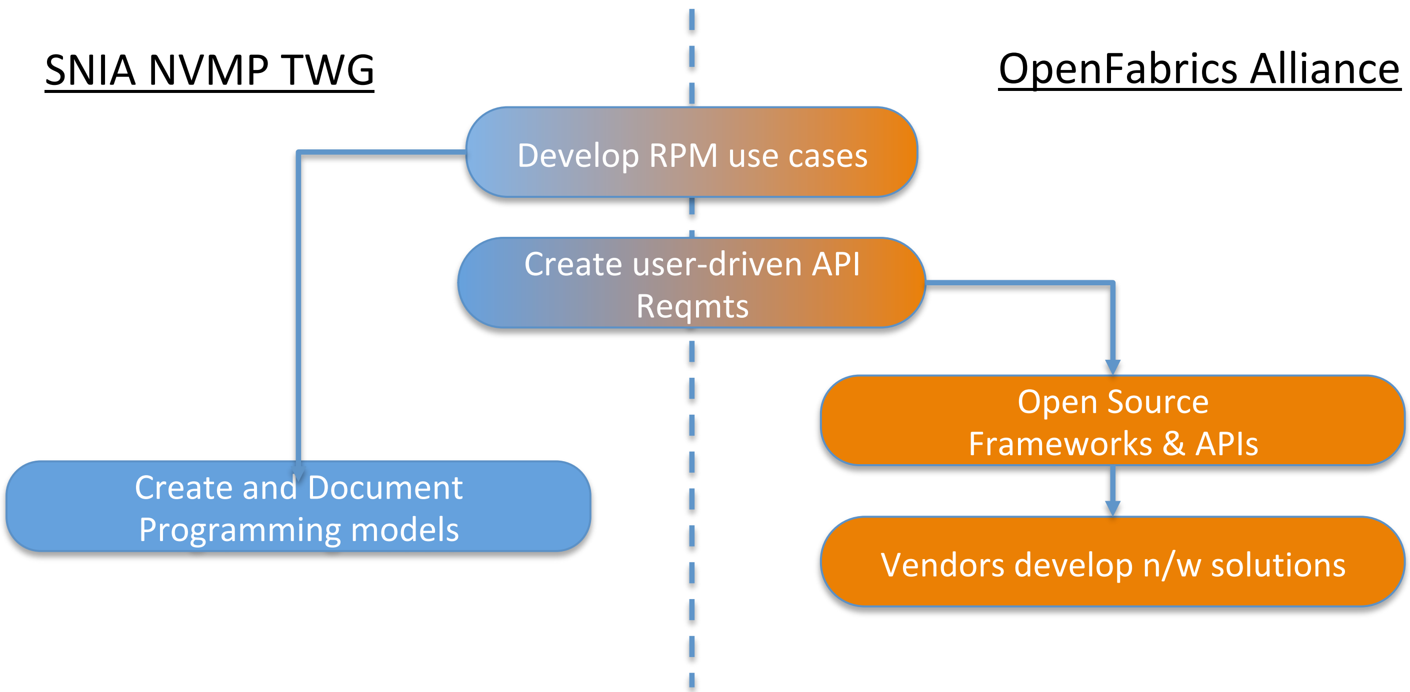
OFA

Both are based on understanding consumers – Application Centric Design

Steps Forward – What's Planned

- Enumerate potential use cases for RPM
  - Use an OFA working group – OFI WG
- Using those use cases
  - Describe new programming models (SNIA)
  - Develop enhancements to network APIs (OFA)
  - Deliver better network solutions (industry)

SNIA/OFA Alliance – How It Works



Brainstorming Use Cases… So Far

1. Local Copy Centric – data is copied from remote PM to local DRAM (or PM) for caching and/or manipulation, then copied back as needed
2. High Availability - Local access to PM + remote access for HA for data recovery and failover with little to no work loss
3. Checkpoint/Restart – Application pauses to enable rapid copy of relevant state to a checkpoint
4. Distributed Collaboration – Remote PM provides a central repository for a distributed team collaborating on a large artifact such a movie
5. Random Byte Range Read After Ingest – Ingest of a large body of data followed by short random reads by parallel threads, e.g. machine learning

Brainstorming Use Cases… so far

6. Aggregated Updates – Cache line accesses such as those comprising a transaction are aggregated for communication to remote PM for visibility and/or redundancy.
7. NUMA on Steroids – Extend and merge the concepts of NUMA, caching, and tiering from CPUs and storage to provide autonomous operation controlled by application informed allocation policies
8. Memory Capacity – Expand memory capacity with lower cost, higher density and larger scale than DRAM
9. Mirrored Transactions – Transactions using local PM are replicated to local PM on other nodes

Brainstorming Use Cases… so far

10. GPU – Copy state directly between GPU memory and RPM without going through DRAM
11. Rehydration – RPM used for DB logs/checkpoints to enable rapid re-hydration of memory after failure
12. Metadata De-amplification – When metadata becomes larger than memory, metadata paging can cause read/write amplification relative to payload data read/write. RPM density can offset this type of amplification.
13. Shared Sensor Data – Streams of information within edge or between edge and centralized repository

Call to Action – Add Your Voice

- Subscribe to the mailing list - Ofa_remotepm
  visit lists.openfabrics.org to subscribe
- SNIA members, participate in the NVM Programming Model TWG
- Join the OFA, Join SNIA

Next Up

- Scott Miller, Dreamworks Animation
  - *Remote Persistent Memory in Feature Animation Production*
- Jim Harrell, Cray, Inc.
  - *HPC and Remote Persistent Memory*
- Idan Burstein, Mellanox
  - *RPM Impacts in Network Architecture*