Excelero

NVMesh™ Customer Profile

NASA – Ames Research Center

NASA Ames Research Center is the home of the NASA Advanced Supercomputing[i] (NAS) facility. NAS enables advances in high-end computing technologies and in modeling and simulation methods to tackle some of the toughest science and engineering challenges facing NASA today. After successfully testing NVMe flash to facilitate interactive database queries on massive data sets of millions of small files, a team at NAS has engaged with Excelero to investigate unifying NVMe at petabyte scale; as a single, shared pool of high-IOPs, low latency storage.

High-end computing and storage

The **High-End Computing Capability Project[ii] (HECC)** provides world-class computing, storage, and associated services to enable scientists and engineers supporting NASA missions to broadly and productively employ large-scale modeling, simulation, analysis and visualization to achieve successful mission outcomes.

HECC technologists provide functional data analysis and visualization software to enhance engineering decision support and scientific discovery by incorporating advanced visualization techniques and displays.

The NASA challenge:
How to leverage the performance of local flash in a distributed fashion without having to worry about data locality.

The Excelero solution:
NVMesh by Excelero enables creation of petabyte-scale unified pools of high-performance flash distributed across multiple servers – yet it retains the speeds and latencies of directly-attached media. It offers the features of centrally-managed block storage, including logical volumes, data protection and failover – with zero performance compromise.

Interactive visualization

In order to support immersive, advanced interactive visualization techniques, NAS developed the hyperwall[iii], one of the largest and most powerful visualization systems in the world. The hyperwall provides a supercomputer-scale environment to visualize and explore the very large, high-dimensional datasets produced by NASA supercomputers and instruments. It helps researchers display, analyze, and study high-dimensional datasets in meaningful ways, allowing the use of different tools, viewpoints, and parameters to display the same data or datasets.

The hyperwall allows researchers to view and interact with tremendously large simulations at a resolution normal displays are incapable of.

hyperwall challenges

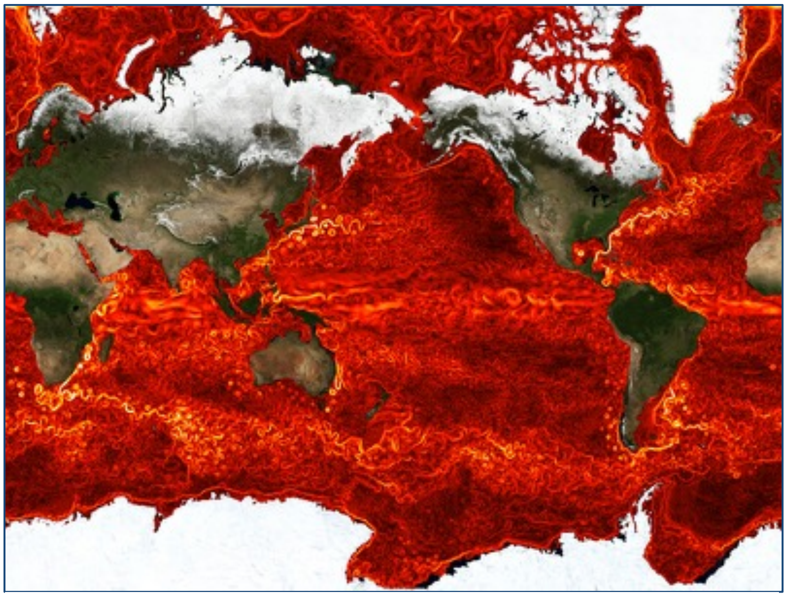
One of the aims of the hyperwall group is to develop an interactive exploratory computing environment. They'd like to do this with bigger and more complicated data sets. Interactive work is generally small IO's; making this a challenging workload with traditional media. They have tried to use the main Lustre file system backed by spinning drives but the small IO, random access patterns generated by their visualization work reduced an 80GB/s file system to only 100's of MB/s of throughput. This behavior is what lead to the decision to introduce NVMe flash into the hyperwall compute nodes.

Flash to the rescue?

In an effort to alleviate random data access issues, the visualization team installed a single 2TB NVMe flash drive in each of the 130 (128 active plus spares) hyperwall compute nodes. In the team's work with SETI[iv] Institute, they do analysis with time varying spectrograms. These data sets are made up millions of small files, in the range of 100KB each. The hope was that this low latency, random access optimized media would help speed things up – and it did, but at a cost: the programmers now had to split the data set up into pieces no larger than 2TB and copy those pieces to each of the 128 compute nodes. Data locality now had to be taken into account during the compute and interactive phases of the visualization process.

Data locality dictates compute methods

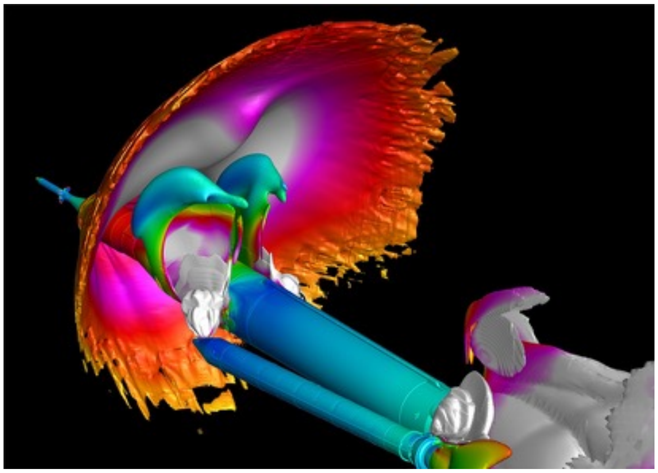
The data locality issue that was introduced by locally deployed flash not only affects the visualization team's work with SETI workflows, but also their work in flow field pathfinding in aerospace simulation, or more recently with projects such as ECCO[v] (Estimating the Circulation and Climate of the Ocean). In the team's work around path integration through flow fields, two techniques are generally applied for computation: in-core and out-of-core. In-core methods are used on data that is in memory or on fast, local media such as flash. Out-of-core techniques are applied when the data to be manipulated is not local to the compute node, or on slow storage. The problem with out-of-core data operation techniques is that they are relatively slow. Life becomes a lot easier for programmers with in-core techniques, which allow programs to access data randomly anywhere in the data set at local speeds.



Surface current speeds from a 1/16-degree solution simulation. Image: Chris Henze, NASA/Ames

Data access affects human interactivity in exploratory computing

When doing interactive exploratory computing, lag in the latency sensitive IO not only affects computing, but also the way human beings interact with the simulation. In the hyperwall team's work around graphical database queries on the SETI database, the team noticed that latency has a significant impact on human interactivity. Humans normally compensate for lag in these demonstrations by slowing down or otherwise altering their behavior. If random access latency to these large data sets was reduced, people could interact with the computing environment in a more natural fashion.



Space Launch System (SLS) booster separation flow field. Stuart Rogers, NASA/Ames

Flash and supercomputing

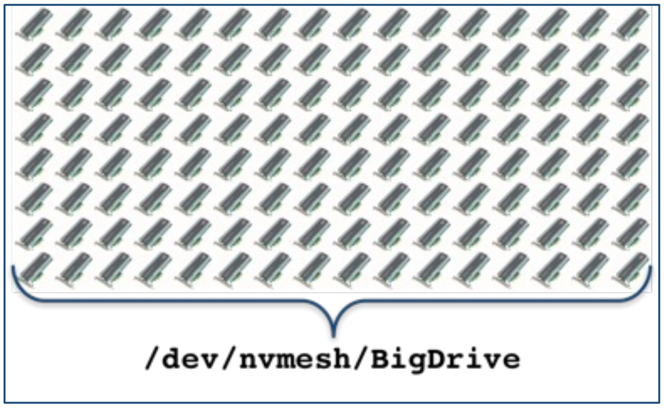
Focusing on low latency, flash seems a perfect fit for supercomputing; however, flash media utilized in centralized storage infrastructure loses much of its latency advantage. Flash media in compute servers increases performance but also increases programming complexity. Hundreds (or thousands) of individual data islands are fast, but they are isolated. Cost and the lack of enabling software have been significant barriers to the adoption of flash memory into supercomputing environments. However, over the next year or so, NAS plans to deploy a significant portion of their storage infrastructure leveraging flash. They want to preserve the performance characteristics of flash memory, aggregate the performance and capacity of 100's or 1000's of devices, and access it through some existing API. Their challenge is how to present that interface to applications that have traditionally relied on standard POSIX file system interfaces. Directly accessing network device targets and leveraging RDMA gives them parallel read access and has advanced their capabilities in the area of data analysis and visualization. It is a good first step towards more complex parallel write arbitration schemes and the software required to do that.

Excelero NVMesh and the hyperwall compute cluster

The visualization group at NASA, Ames Research Center has purchased and installed NVMesh on the hyperwall compute nodes. NVMesh by Excelero creates a petabyte-scale shared NVMe pool across multiple servers – and makes it available over a network at speeds and latencies of directly-attached media. It offers the features of centrally-managed block storage, including logical volumes, data protection and failover – with zero performance compromise. Combining blistering speeds with software-defined flexibility, NVMesh takes storage to its next generation, today. It transforms the performance, the economics and even the feasibility of multiple use cases in visualization, analytics/simulation, and burst buffer.

Data Locality: make ALL data appear to be local

The first use case on the hyperwall was to aggregate the distinct 2TB flash drives into a single logical device. This logical block device served as the basis of a file system to be accessed by all 128 of the compute nodes. Because the simulation data is protected in the main Lustre file system, this single large device was to be treated as ephemeral. It was created without data protection as a RAID-0 logical volume striped across all 128 nodes/devices. When attached to all 128 compute nodes, it appeared and behaved as a single 256TB block flash device. For simplicity, the device was attached to a single node, formatted with an XFS file system and populated with data. The file system was then unmounted and mounted (read-only) on all 128 compute nodes. While a local file system was used for simplicity, NVMesh logical block volumes can be used with clustered file systems and can also be protected against host or drive failures.



NVMesh takes flash devices distributed over 100's of hosts and unifies them while preserving local flash performance

Ultra-high performance and scalability

NVMesh adds a mere 5µsec latency (mostly from network latency) over local NVMe drive latency. This means that any compute node in the visualization cluster can access any logical block in the 256TB file system in as little as 85µsec. Applications can be written (and they behave) as if they have a massive, locally attached flash drive. There is no need to plan for nor worry about if data is local to a node – all data accesses behave as if they are local to every compute node.

Utilizing the configuration described above, with fio[vi] benchmarks running on all 128 compute nodes, preliminary results have demonstrated over **30 million** random read 4K IOPs. **The average latency for those IOPs was 199µsec**. Throughput has been measured at over **140 GB/s** of bandwidth (at 1MB block size).

NVMesh allows logical aggregation of distinct flash devices into a unified block pool – allowing for access methods such as direct mapping or POSIX with your choice of file system. It does this while preserving the latency and performance characteristics of locally installed flash, making this an optimal choice for deployment of flash in supercomputing and HCP environments. Lastly, when utilizing native NVMe queuing mechanisms **it completely bypasses CPU on the target hosts** (those with NVMe drives) preserving processing power for applications.

The end result

Visual demonstrations on the hyperwall run more smoothly with notably less lag. Human beings can interact with visual, simulated environments in a more natural fashion. Programmers have the freedom to treat the entire data set as if it is local to every node. Compute nodes can access data anywhere within a data set without worrying about locality – because the entire data set is accessible randomly at low latency and high bandwidth.

Excelero

NVMesh™ Customer Profile

Excelero NVMesh

NVMesh is a 100% software solution. Customers can choose servers, networks and NVMe media and combine them to tailor solutions with the latency, bandwidth and endurance parameters that fit their needs. While there are Excelero-certified components from leading suppliers, customers can mix vendors, drive types and sizes - truly commoditizing their storage infrastructure. They can also grow the system easily, simply and cheaply.

Deployment: Converged or Disaggregated

NVMesh is available in 2 deployment models, both combining ultra-low latencies with the benefits of a centrally managed, shared storage pool:

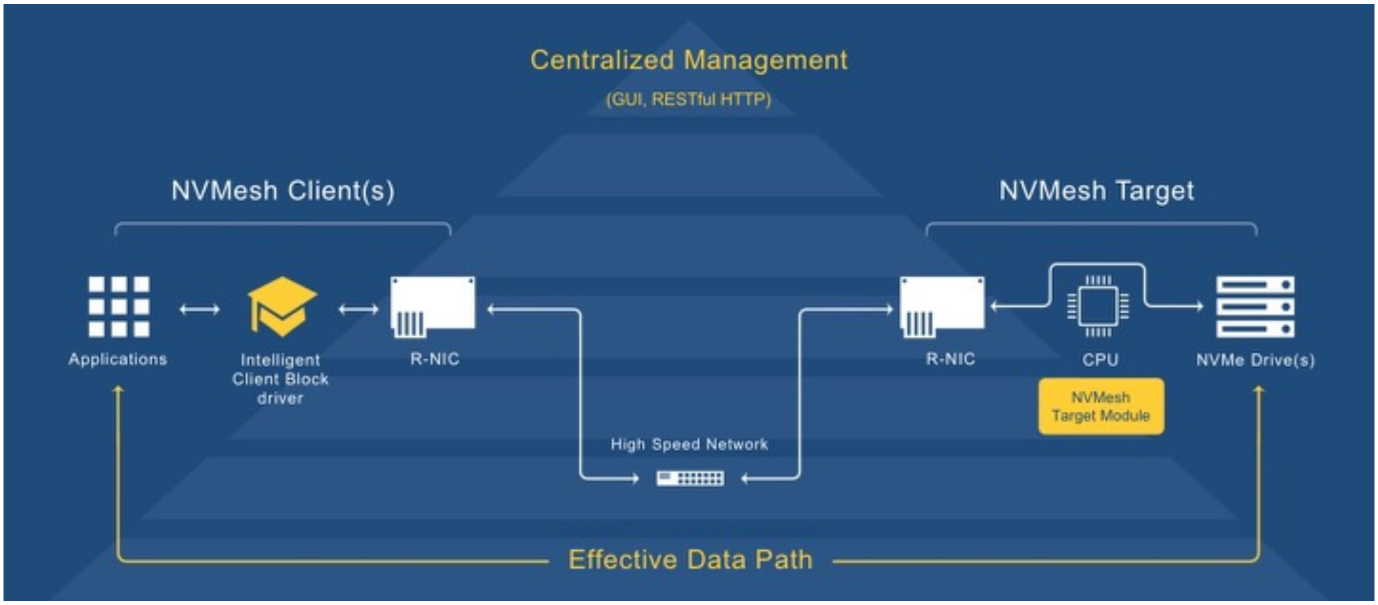
Converged deployments offer the highest levels of aggregate performance and the lowest TCO. Uniquely, NVMesh enables true convergence in that **it does not consume CPU on the target**; removing a major bottleneck, boosting performance and enabling virtually unlimited scalability.

Disaggregated deployments allow for independent scaling of compute resources, with high operational efficiency and serviceability. They allow for scaling capacity at ultra-low latencies for the largest, most demanding data sets that are not bandwidth bound.

Architecture and components

NVMesh comprises 3 software modules:

- The **Storage Management Module** is a centralized web-based GUI, RESTful API that controls system configuration. It also interfaces with the Docker Persistent Volume plugin and OpenStack's Cinder drivers.
- A **Target Module** is installed on any host that shares its NVMe drives. It validates initial connections from clients to the drives, but keeps out of the data path.
- A **Client Block Driver** runs on every host/image that wants to access NVMesh's logical block volumes. In converged deployments, the Client and Target Modules coexist on the same server.



[i] NASA Advanced Supercomputing (NAS): https://www.nas.nasa.gov/about/about.html
[ii] NASA High-End Computing Capability Project: https://www.nas.nasa.gov/hecc/about/hecc_project.html
[iii] hyperwall: https://www.nas.nasa.gov/hecc/resources/viz_systems.html
[iv] SETI Institute: http://www.seti.org/
[v] ECCO: http://www.ecco-group.org/index.htm
[vi] Flexible I/O Tester (fio): https://github.com/axboe/fio