TECHNICAL WHITE PAPER

Scaling EDA Workloads on Azure and Pure Storage in Hybrid Cloud

A modern data platform experience for scaling in performance and capacity on demand.

Contents

Scaling EDA Workloads on Azure and Pure Storage in Hybrid Cloud

Electronic design automation (EDA) tools are used in the semiconductor chip design, simulation, and manufacturing work that requires a large amount of parallel computation and storage that can operate at scale. The chip design dataset primarily consists of millions of small files that have concurrent file access from hundreds of nodes. EDA tools can access different parts of the same or more files in parallel for read, write, and other metadata operations.

The workload profile of semiconductor chip design and modeling varies from high metadata to high bandwidth, with a requirement for acceptable latency, and the ability to perform read and write operations to one or many files in parallel. Vertical scaling with several compute nodes in clusters requires shared storage to process and analyze existing data or generate new data.

With the introduction of smaller chip sizes used in mobile devices, the silicon on chip (SoC) form factor is diminishing, with a higher yield rate requirement. Designing and manufacturing the new-generation SoC requires highly scalable compute and storage that crosses data center boundaries and extends to hyperscalers with the flexibility to burst into the cloud on-demand. Azure cloud provides virtual machines (VMs) as compute requirements for EDA tools with a high degree of scalability across multiple Azure regions.

Many organizations connect to the Azure cloud from their respective data centers using Azure ExpressRoute for cloud bursting, where the data is local to the business owners. However, a growing number of EDA and semiconductor business owners want to move away from a CAPEX model of paying upfront costs on infrastructure and its maintenance while maintaining strong security and control over their data. Intellectual property and chip design data is an integral part of the design and tapeout workflow. Moving data into the cloud leads to data sovereignty issues, including security and legal ramifications.

In highly parallelized workloads like EDA and semiconductor design and manufacturing, there is a strong need to disaggregate the data platform from the cloud compute, where storage can be consumed on-demand as an OPEX model without any upfront cost. Pure Storage® FlashBlade® provides a modern data platform experience that can scale in performance and capacity on demand. FlashBlade can be managed in connected cloud data centers like Equinix and connected to Azure VMs using ExpressRoute. The following figure illustrates the connectivity of Azure VMs to FlashBlade in Equinix using Equinix Cloud Exchange (ECX) Fabric.

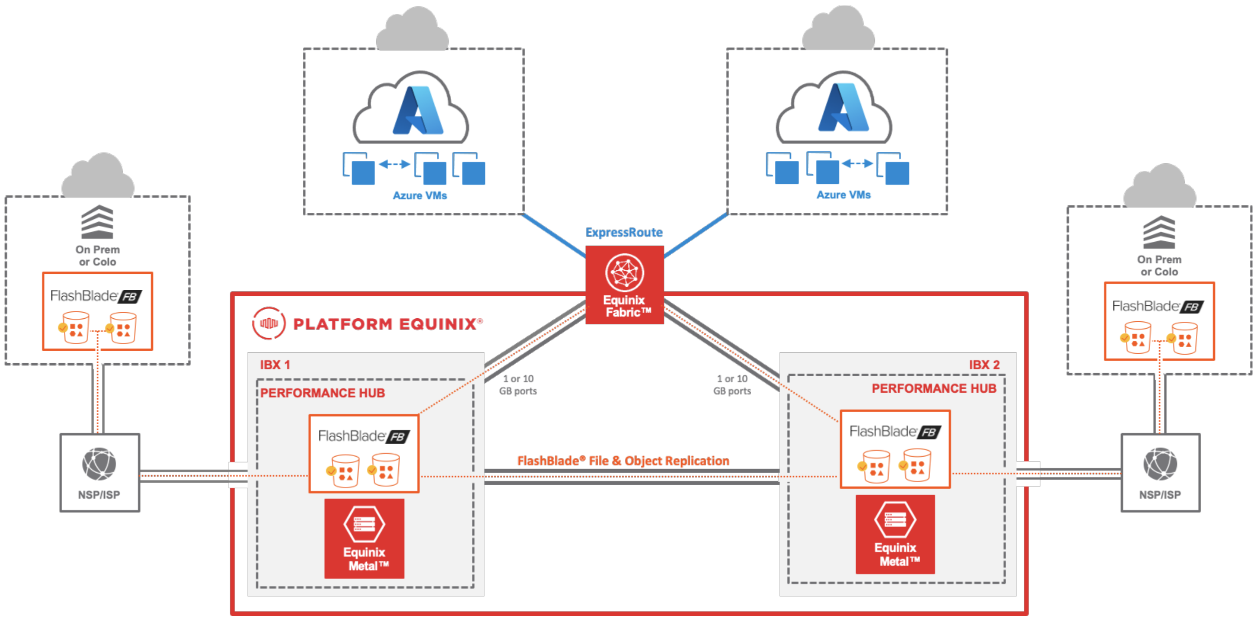


Figure 1. High-performance interconnected storage

FlashBlade provides the flexibility to use different data access protocols like NFSv3/4.1, SMB, and S3 on the same platform. Programmable infrastructure using Ansible playbooks allows for automated provisioning of storage on FlashBlade. Ansible playbooks also provide automated and controlled data mobility between the FlashBlade system on-premises and in cloud-connected data centers for staging before bursting into Azure cloud. Data is secure on-premises and in Equinix locations as FlashBlade natively provides encryption for data at rest.

Why the Hybrid Cloud Approach?

Azure cloud integration with FlashBlade in a cloud-connected data center becomes a compelling choice for many business owners for the following reasons:

- **Data security and control**: FlashBlade systems in cloud-connected data centers are customer-owned or fully isolated to single-tenant customers through services such as Pure Storage on Equinix Metal. This ensures the control and security of the data in a cloud-connected data center outside of the on-premises data center with complete knowledge of the data location. Data governance, mobility, and protection can be customized to specific business needs. Moving the data to a connected cloud data center allows business owners to connect to any major public cloud provider.
- **Simplify cloud adoption**: Automating the data mobility from the FlashBlade system by using Ansible playbooks between on-premises and thecloud-connected data center location provides a zero storage touch to enable the end-users to onboard with any cloud provider on-demand. Rapidly staging data in the cloud-connected data center location allows business owners more freedom to scale the compute resources to host customers or perform cloud bursting.
- **Performance at scale**: The chip design signature workloads can scale compute and storage independently on-demand to maintain a linear performance growth with respect to capacity and cost.
- **Cost efficient**: Long-running chip design and tapeout jobs can finish quickly, thus reducing the cloud compute and egress costs. Faster job completion also accelerates the entire chip design workflow, thus optimizing the EDA tool license costs. Data reduction on FlashBlade further reduces the data footprint, allowing more data on less storage.

This white paper will highlight two areas of focus for EDA and semiconductor design workloads:

1. Performance evaluation on a validated architecture with Azure VMs and FlashBlade in a cloud-connected data center using:
    a. EDA workloads using SpecStorage2020 benchmark
    b. Scaling open source Linux kernel builds
    c. Hasbro tests using VDbench scripts
    d. Spectre X Simulation tool from Cadence
2. Data mobility automation between on-premises and cloud-connected data centers with improved performance and security

Performance Validation with Azure Cloud and FlashBlade in a Cloud-connected Data Center

The data platform performance for SoC workloads in the recent past was primarily measured by the maximum throughput it could deliver at peak load from large arrays of mechanical disks. More recently, EDA tool workload profiles have become focused on smaller files and IOPs, and they require high-performance storage that can meet these needs with low latency. Certain flash-based architectures are potentially well suited for these requirements:

IOPs/sec: A unit of measurement for flash-based data platforms that validates the rate of data transfer rate based on the I/O requests from the application. Higher IOPs/sec (or IOPS) from a flash-based data platform could be a good indicator of handling SoC workloads at scale. However, a high number of IOPs served from the data platform is incomplete without measuring latency. Data platforms with high IOPs and high latency at scale will not be a good solution for SoC workloads.

Latency: A unit of measurement for how fast the I/O requests from the application are handled by the data platform. In a file-based shared data platform, latency could be attributed to various layers between the application and the data platform. A roundtrip time of an I/O from the application to the data platform and back is a good measure. However, latency can be dependent on various components of the data path, such as:

- Concurrency with a single thread vs. multithreading from each node in the cluster
- Linux nodes that overload the send and receive socket (TCP) buffers or aggressively swap to backend storage, at scale
- RPC calls (delayed TCP ACKs, retransmissions) and network topology
- Data platform speed and headroom available to process the I/O requests

Throughput: A measurement of the transfer speed of the I/O requests from the application to the data platform in MB/sec. Throughput depends on IOPs and the average NFS block size or I/O size. NFS block size refers to the block size that the application uses and the negotiated NFS transfer size (rsize/wsize for NFS mounts) between the Linux client and the backend data platform.

Concurrency: Defines the parallel I/O requests processed by the data platform at any given point in time. Concurrency plays a major role for distributed and EDA tools at scale. With linear scaling, the data platform should function in this complete efficiency by processing as many parallel I/O requests with less or no pending I/O in the queue.

Little's Law states the relationship between concurrency, latency, and throughput. A single stream of I/O with low latency will not generate a high throughput without a certain degree of concurrency.

FlashBlade

Pure Storage FlashBlade provides a direct flash data platform for standard NAS and SoC workloads that allows for performance (IOPS and bandwidth) and capacity scaling independent of the compute layer. FlashBlade supports many integrated virtual network interfaces in a single namespace along with compute, memory, and flash storage on each blade. The global lock distribution and parallel metadata and data access using non-overlapping partitions to control and manage data provide low latency to the EDA tools at scale from a data platform perspective.

All-flash arrays from Pure Storage have already redefined performance at scale concerning IOPs, bandwidth (GB/sec), and latency (ms). This paper highlights some of the key performance indicators for different workload profiles that are most found in semiconductor chip design tools and manufacturers.

Reading and writing a large number of small or large files from deep directory structures generate massive amounts of metadata and bandwidth. FlashBlade has been designed to handle these kinds of high metadata- and throughput-based workloads that are common in EDA/chip development environments.

The primary objective of the scalability test was to measure the following metrics from the FlashBlade system as more blades were added to the data platform:

1. Measure the op_rate (IOPs/sec) with respect to latency using the SpecStorage2020 benchmark.
2. Measure the completion time for parallel builds with an open source Linux kernel.
3. Measuring read/write throughput/IOPs and metadata IOPs at scale with Hasbro tests.
4. Validating job completion time with the Spectre X tool from Cadence.

SpecStorage2020

The Azure VMs mounted the "specsfs2020" file system from the FlashBlade system over NFS. The SpecStorage2020 tests were performed on a single Azure-E64d_v4-VM over an Ultra Performance gateway with Azure ExpressRoute speed of 10Gb/sec from FlashBlade in the Equinix cloud-connected data center as shown in Figure 2. The 45-blade FlashBlade platform used for the performance validation had a 52TB capacity blade.

The default workload definition in SpecStorage2020, "EDA_Blended", was used to test the workload from the Azure VM with four NFS mounts to provide I/O parallelism to the FlashBlade system. The tests were performed over NFSv3 and NFSv4.1 protocols respectively, with the recommended NFS mount and kernel optimizations on the Linux host.

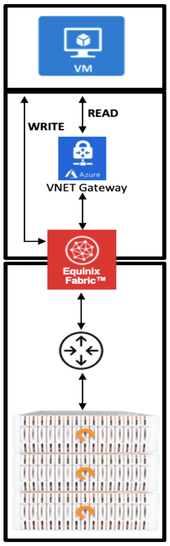


Figure 2. Performance test layout

Four working directories were specified in the CLIENT_MOUNTPOINTS parameter in the Specstorage2020 configuration file to distribute the EDA_BLENDED workload definition that includes the front-end and back-end module. The front-end module simulates high metadata operation normally generated during build cycles during logical verification.

```
...
CLIENT_MOUNTPOINTS=azpr-pureraymond:/mnt/specsfs2020-1 \
       azpr-pureraymond:/mnt/specsfs2020-2 \
       azpr-pureraymond:/mnt/specsfs2020-3 \
       azpr-pureraymond:/mnt/specsfs2020-4
...
BENCHMARK=EDA_BLENDED
…
```

The back-end definition includes a metadata and high bandwidth workload that is seen during simulation phases of the chip design process. The FlashBlade system in the Equinix cloud-connected data center was observed to have a 2ms latency round-trip time from the Azure VMs.

The following NFS mount options were used on the Azure VM for each of the four mounts for running the SpecStorage2020 EDA_BLENDED definition.

> NOTE: The following mount options were used for all performance and functional tests for EDA tools and benchmarks performed on the FlashBlade system.

```
192.168.100.101:/specsfs2020      /mnt/specsfs2020-1        nfs
hard,rw,bg,vers=3,nolock,tcp,timeo=600,actimeo=600
192.168.100.102:/specsfs2020      /mnt/specsfs2020-2        nfs
hard,rw,bg,vers=3,nolock,tcp,timeo=600,actimeo=600
192.168.100.103:/specsfs2020      /mnt/specsfs2020-3        nfs
hard,rw,bg,vers=3,nolock,tcp,timeo=600,actimeo=600
```

```
192.168.100.104:/specsfs2020      /mnt/specsfs2020-4        nfs
hard,rw,bg,vers=3,nolock,tcp,timeo=600,actimeo=600
```

The following kernel parameters are recommended to be configured on the Linux hosts used in various performance validation tests mentioned in this document:

```
cat /etc/sysctl.conf

fs.file-max = 2097152
net.core.somaxconn = 1024
net.core.netdev_max_backlog = 300000
net.core.optmem_max = 16777216
net.core.wmem_max=16777216
net.core.rmem_max=16777216
net.ipv4.tcp_rmem= 4096 87380 16777216
net.ipv4.tcp_wmem= 4096 87380 16777216
net.core.rmem_default = 16777216
net.core.wmem_default = 16777216

cat /etc/security/limits.conf
root             -       nofile          10000

cat /etc/modprobe.d/sunrpc.conf
options tcp_slot_table_entries=128 tcp_max_slot_table_entries=128
```

NOTE: Some of the kernel tuning parameters may not apply for newer Linux kernels 4.12 and later. “nconnect” was not used for this performance validation with Specsfs2020 over NFSv3 and NFSv4.1

Observations

The SpecStorage2020 test was performed with Azure VM using a Standard gateway and compared with an Ultra-Performance gateway over NFSv3. The Azure VM using the Standard gateway generated 27,000 operations per second at a latency of 2ms, beyond which the latency started to increase. The Azure VM with an Ultra Performance gateway generated 54,000 operations per second at 2ms latency.

The operations per second doubles and the “knee” of the latency to operations per second curve moves further to the right as shown in the figure below, demonstrating linear scalability at low latencies while using the Ultra Performance gateway. The test indicates a moderate increase in latency as the number of operations per second scales, thus indicating that the network bandwidth is saturated. The FlashBlade system has enough headroom to accommodate more workloads as the Azure compute scales independently, until the network limitations are met.

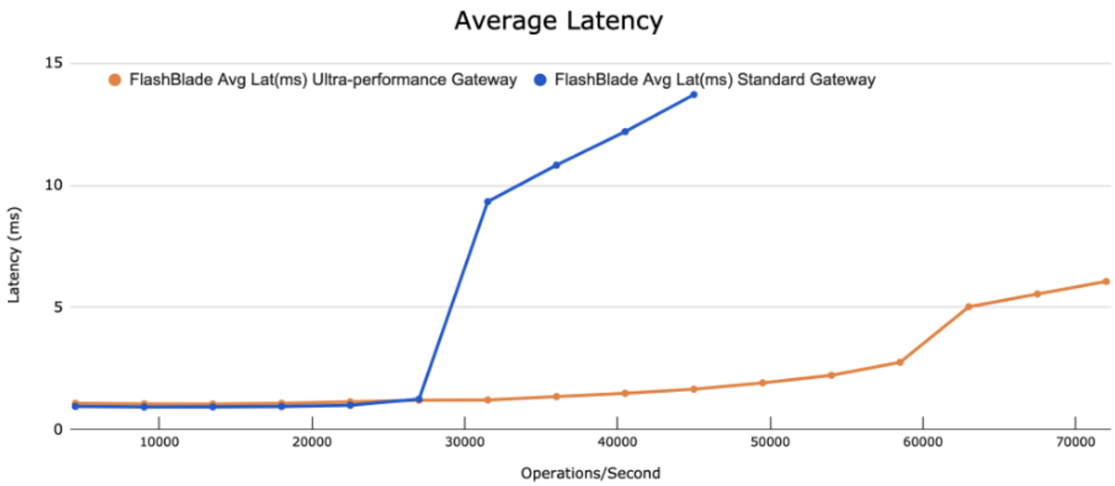


Figure 3. SpecStorage2020 ops/sec vs. latency graph

Similarly, the SpecStorage2020 EDA_BLENDED test was performed using an Azure VM and an Ultra Performance gateway to FlashBlade over NFSv4.1. The following figure shows the NFSv4.1 operations per second and latency comparison with NFSv3. The test results indicate that NFSv4.1 was able to generate 40,000 operations per second at 2ms latency. The NFSv4.1 operations per sec and the latency curve are not at parity with NFSv3. However, the knee of the curve for NFSv4.1 is not too far behind compared with NFSv3.

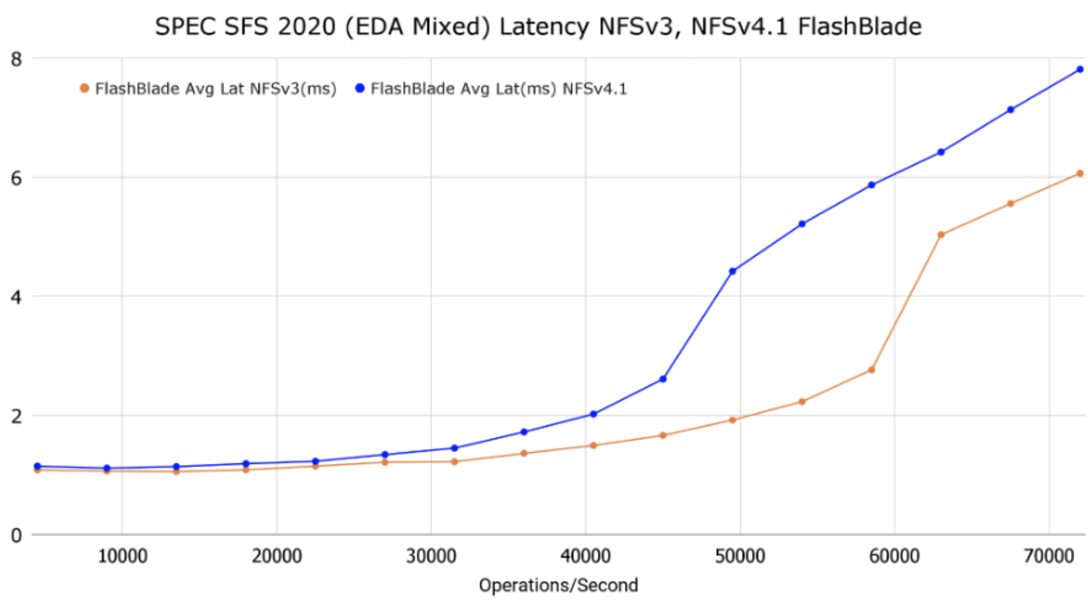


Figure 4. SpecStorage2020 latency vs. NFSv3/ NFSv4.1 graph

The test results proved to be encouraging and contradict the poor performance of a 2ms round-trip time between the Azure VM and FlashBlade in the Equinix cloud-connected data center. For EDA workloads, an E64dsv4 Azure VM with an Ultra Performance gateway and 10Gbps ExpressRoute connection to the FlashBlade system scales the IOPS linearly and consistently with a latency of under 2ms.

Parallel Builds

Incremental and parallel builds are very common during the logic simulation and dynamic testing phases of SoC design. It is not uncommon for pre-commit smoke tests and mini-regressions to scale to many thousands of tests, with full regressions being orders of magnitude larger. The builds require predictable regression completion time as the number of parallel builds scale over time. While "makefile" generates a lot of I/O operations to the backend storage, not all storage can deliver consistent completion time at scale.

An open-source Linux kernel was used for the purpose of validating the parallel build on FlashBlade and measuring the job completion time as the number of builds scaled from using a single VM to 15 VMs in the Azure cloud. Incidentally, this test is an excellent benchmark for testing storage performance for small file and metadata-dominant workloads that we see very commonly in software development, EDA and semiconductor chip design, artificial intelligence, deep learning, animation and CGI rendering, genomics, and fintech workloads, just to name a few. The Azure VMs connect to the FlashBlade in the Equinix location over ExpressRoute with around-trip time of 2ms.

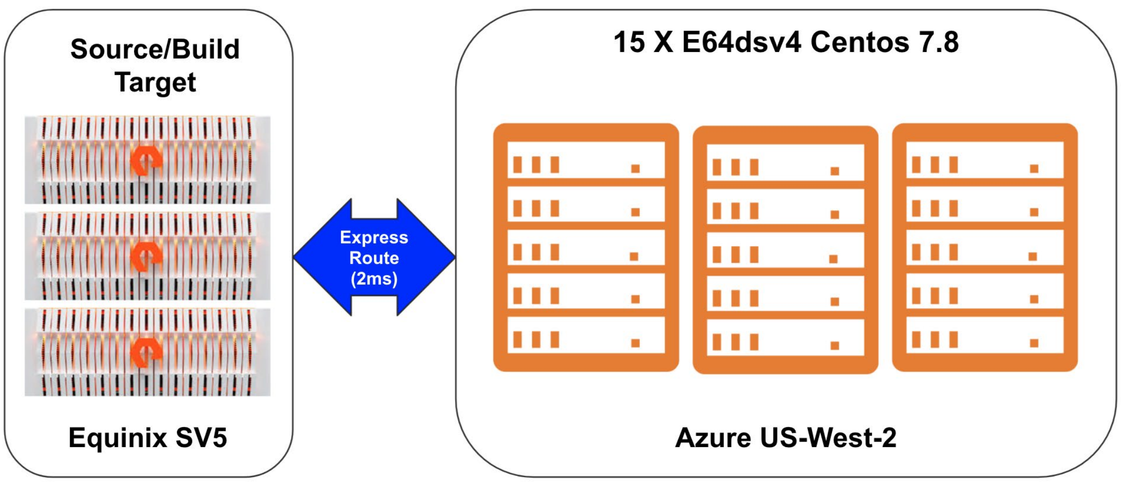


Figure 5. Parallel build test environment

Two different build experiments were run:

1. Two threads per build, with eight builds per instance, scaling up to a total of 120 parallel builds (8 builds/instance * 15 instances).
2. Eight threads per build, with three builds per instance, scaling up to a total of 45 parallel builds (3 builds/instance * 15 instances). The three build types were run, each with a different architecture target (i86, x86, and ARM).

Observations

Both build experiments showed exceptional results. The ideal situation in all cases of a parallel software build is that storage saturation does not cause execution time to go up as load increases, and this is exactly what was observed.

For the experiment with two threads per build, the results are graphed below, and it clearly shows that from 8 to 120 builds there is practically no impact to build runtime.

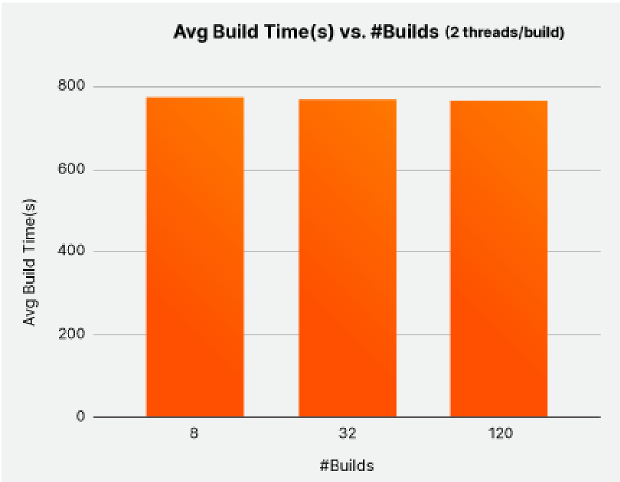


Figure 6. Parallel build time vs.number of builds

The FlashBlade dashboard showed a very linear response during the test. There are a few other observations that can be made based on the dashboard data.

- The workload is IOPS intensive, and the IOPS ramp up to approximately 135K IOPS is very linear.
- Latency is consistently sub-2ms.
- Bandwidth peaks are ~45MB/s.

Given these data points obtained during this test, it is possible the FlashBlade system could support well over 1,200 parallel Linux kernel builds without any runtime degradation before the ExpressRoute bandwidth limit would start to impact performance.

Figure 7. Metadata IOPS scale linearly with the number of builds

The eight-thread build experiment showed very similar results, with build times practically unchanged as the number of parallel builds were scaled up to 15 Azure VMs as shown in the Y-axis of the following figure.

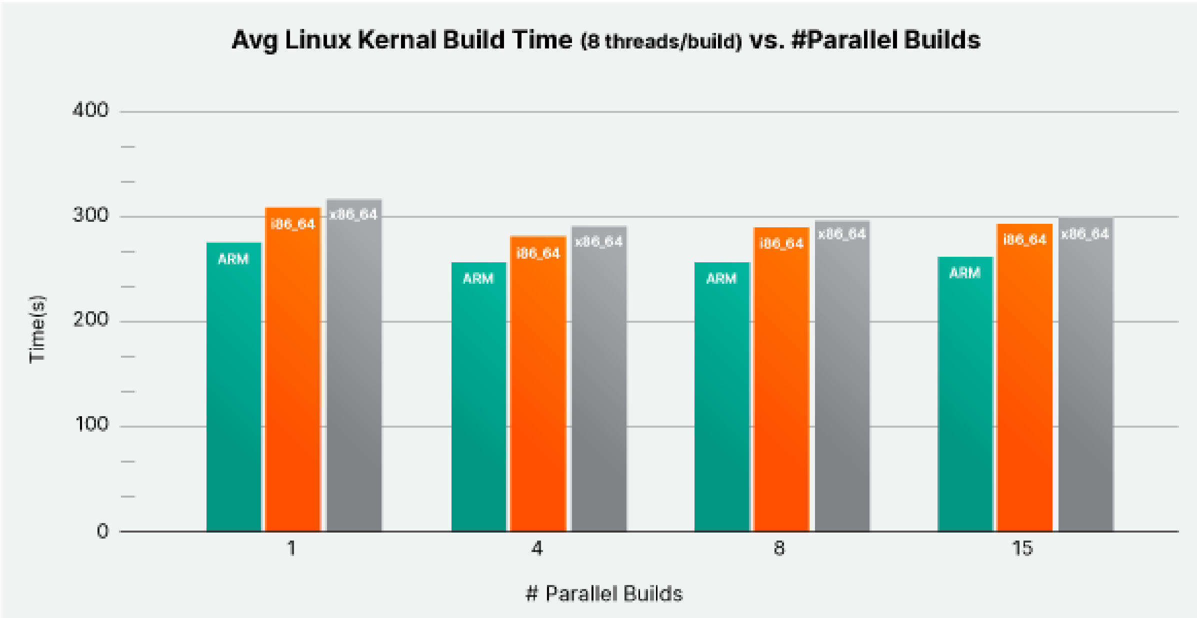


Figure 8. Build time for different CPU architectures at scale

From the above result, it is quite evident that the parallel builds for each of the CPU architecture tests have a consistent build completion time at scale.

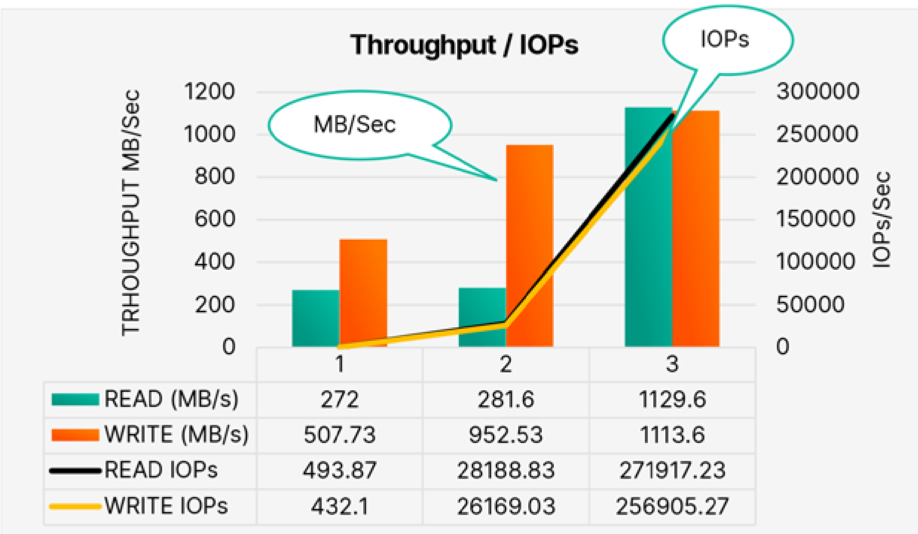
Azure HPC Broad Spectrum Tests

The HPC tests by Azure were performed using VDbench configurations to simulate load for various application profiles. Targeted workloads include AI training and inference, computational chemistry, and weather simulation. All tests reported were run with a single thread per file. Concurrency is probed with 32MiB chunks up to six threads per host, 16KiB chunks up to 64 threads per host, or a metadata load of up to 92 threads per host. These processes were then scaled out up to 24 hosts. Hosts are VMs, each with 2GiB/s networking limits using Azure accelerated networking and the POSIX API. Two broad-spectrum tests were performed in two parts to validate the linearity as the following workloads scaled from single thread to 24 VMs:

1. Read/write throughput and IOPS for each of the three scenarios mentioned above.
2. Metadata IOPS like GETATTR and SETATTR operations for each of the test scenarios. This workload was doing 92 threads per node with one thread per file.

Observation

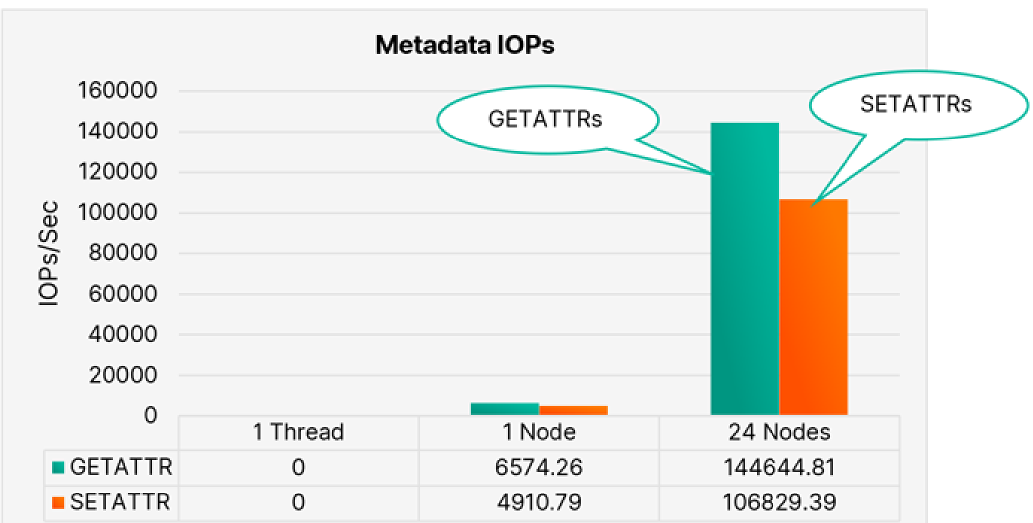
The read/write and IOPS metrics demonstrated complete linear growth as the workload scaled from a single thread to 24 VMs. During the tests, the FlashBlade system demonstrated linearity, with enough headroom to allow more read and write with IOPS workloads as it scaled.



	1	2	3
READ (MB/s)	272	281.6	1129.6
WRITE (MB/s)	507.73	952.53	1113.6
READ IOPs	493.87	28188.83	271917.23
WRITE IOPs	432.1	26169.03	256905.27

Figure 9. Read/Write throughput and IOPs/sec at scale

The metadata IOPS tests for the above-mentioned workloads from single thread to 24 VMs also showed linear scale, with very little stress to the FlashBlade system. The FlashBlade system is capable of scaling to many million metadata IOPS at scale.



	1 Thread	1 Node	24 Nodes
GETATTR	0	6574.26	144644.81
SETATTR	0	4910.79	106829.39

Figure 10. Metadata IOPS at scale

The Azure HPC Broad Spectrum tests also include testing with a parallel I/O application, h5perf using MPI-IO API. This test is representative of parallel HPC workloads where concurrency is many threads on remote systems interacting with different sectors of the same file simultaneously. Using h5perf with up to 24 VMs, with 24 concurrent threads on each host, all to a single file, we measured 1.08GB/s write throughput and 0.95GB/s read throughput—equivalent to the single thread per file pattern.

Spectre X Logic Simulation Tool

Spectre X from Cadence is one of the most popular parallel circuit simulators used during the chip design process. The simulation data storage directory was configured on the FlashBlade system, and the real tests were performed with 1, 2, 4, 8, and 16 threads from the Azure VM over NFSv3. The following test results show a steady decrease in the job completion time as the number of threads increased.

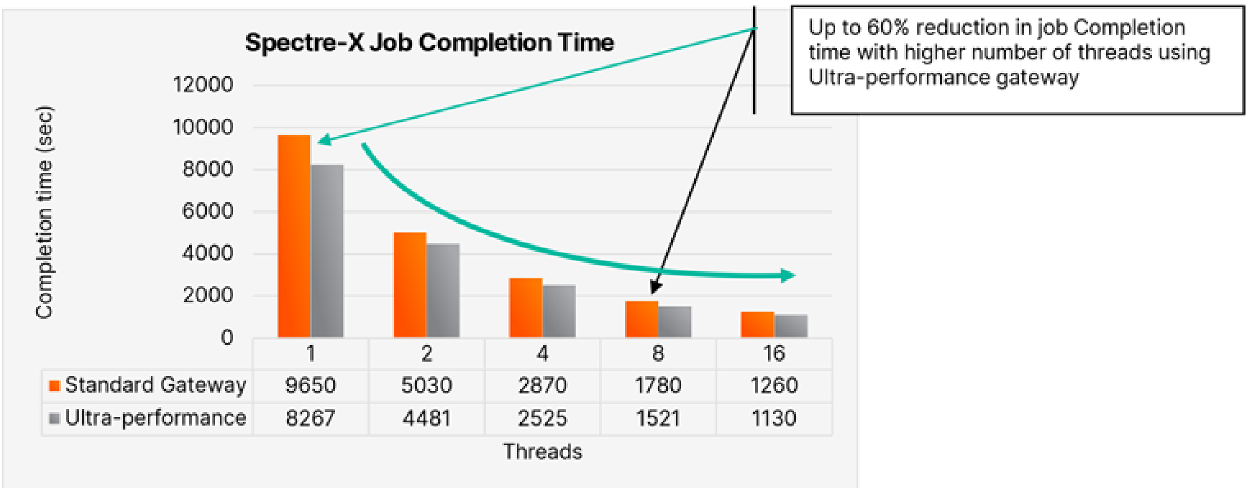


Figure 11. Linear reduction in job completion time with a higher number of threads

The above graph shows up to a 60% reduction in job completion time with eight threads compared to the four threads setting. Depending on the Linux hosts and the network bandwidth, it is recommended to start with eight threads while accessing the NFS simulation data file system on FlashBlade. Using the Ultra Performance gateway with the 10Gbe/sec ExpressRoute, the Spectre X simulation tool was able to perform up to 15% better compared to the Standard gateway with multithread testing until network saturation was met.

Data Mobility in Hybrid Cloud

The paper so far focuses mainly on the performance and functional aspects of the workloads generated from different EDA tools. It is imperative to move the data from on-premises to Equinix location to stage on-demand before bursting Azure cloud. The data mobility is possible using file system replication features from the on-premises FlashBlade to the FlashBlade in Equinix.

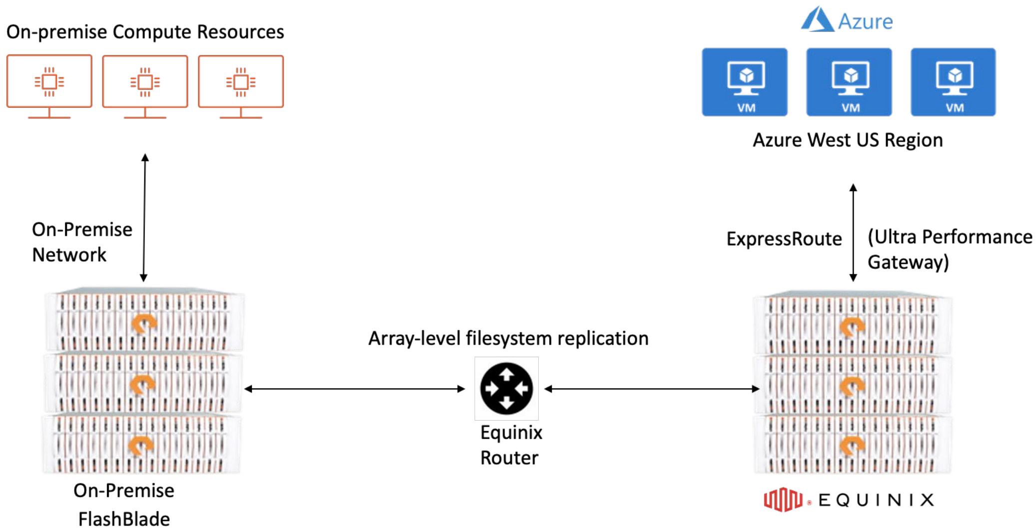


Figure 12. Data mobility between on-premises and Equinix using file system replication

The replication target on the FlashBlade system in Equinix is always a read-only file system. However, the target file system can be promoted to read and write for the Azure VMs to mount that file system over NFS. Meanwhile, the original source file system on the FlashBlade system in the on-premises location is demoted to a read-only state. Promoting and demoting the replicated file systems on both FlashBlade systems allows administrator control over data access.

The file system replication between the on-premises and Equinix-based FlashBlade systems are tunneled through the Equinix virtual router without exposing it to any external network or Azure VMs. This provides security while moving data from on-premises to Equinix and vice versa. Automating Ansible playbooks allows end-users to configure file system replication, and to promote or demote a filesystem, with zero storage touch. The sample file system representing chip design workload demonstrated an average data reduction of 2:1 on both FlashBlade systems. The sample dataset with a 2.3TB size was replicated to the target at the speed of 8.5GB/sec. The ability to have SafeMode™ snapshots on FlashBlade on-premises and in Equinix provides enhanced protection against ransomware.

Data Tiering in Azure Cloud

Data mobility from on-premises FlashBlade to cloud adjacent locations like Equinix using Pure Storage file system replication was explained in an earlier section. File system replication allows the staging of the data in the FlashBlade system on Equinix to use the compute resources on-demand in the Azure cloud for cloud bursting.

However, after the life of the project, you may choose to move the data back on-premises or need to move the data from the FlashBlade system in Equinix to cold storage, like Azure Blob storage in Azure, for long-term retention. While array-level replication can be used to move data back from theFlashBlade system in Equinix into the local data center; host-based tools like fpsync can be useful to move data from an NFS share on FlashBlade to blob storage in Azure.

Fpsync is a powerful open-source migration tool that uses fpart and rsync to migrate small and large files across heterogeneous storage endpoints and data formats. While fpart synchronizes directories in parallel, rsync does the data copy from the source to the target locations. The fpsync tool has a faster file transfer rate irrespective of the file sizes and the size of data that is copied compared to the standard UNIX cp tool and other open-source tools available.

A typical semiconductor chip design environment has high file counts with a deep directory structure and millions of small files with soft and hard links. Fpsync is a very effective host-based tool to migrate design and simulation data across heterogeneous data platforms.

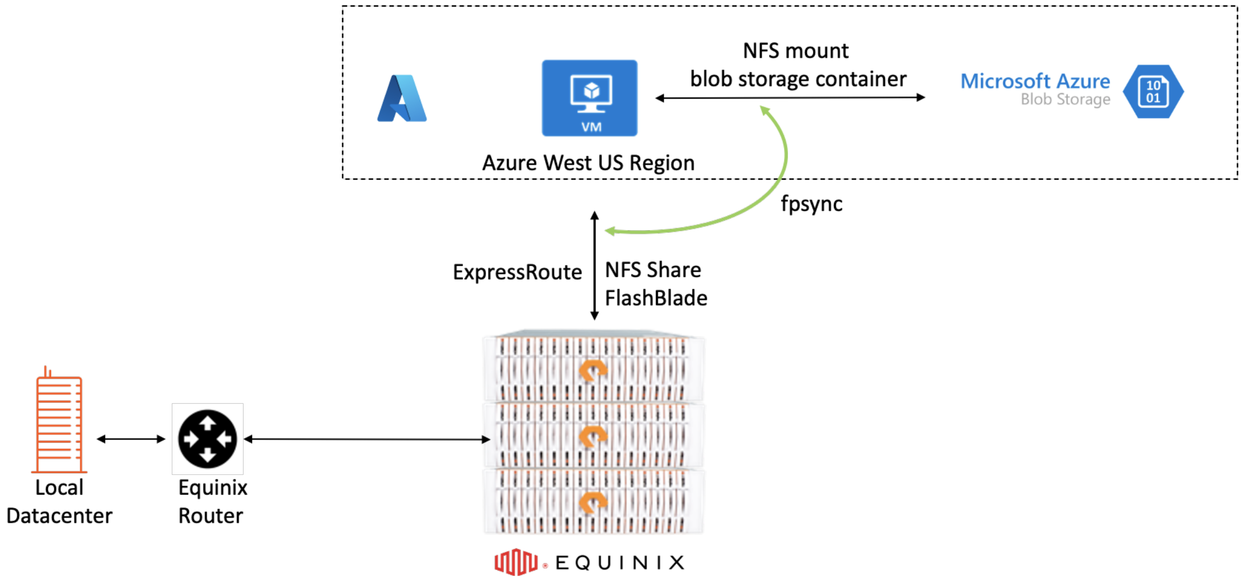


Figure 13. Data tiering from FlashBlade in Equinix to Azure Blob storage

As shown in the above diagram, there may be a lot of residual data on the FlashBlade system from the design and simulation workflow in the Equinix location that may not be required to be retained on primary high-performance storage. The business owner may decide to retain the data in cheap and deep blob storage in Azure cloud.

Azure provides a hierarchical namespace in Azure Data Lake Storage Gen2 that requires arranging the blob with a valid storage account. This feature provides the admins and end-users with the following two main advantages:

1. It arranges the objects in the object store bucket as files and directories.
2. It allows the blob to be mounted on the compute host as an NFS share with a mount path.

Data tiering from FlashBlade in Equinix into blob storage in Azure cloud allows you to archive the data for long-term retention. Using fpsync, data can be moved in both directions—from FlashBlade to Azure Blob storage and restored on demand. Reading from blob storage becomes easy with hierarchical namespace and the mount path over NFS because the objects are arranged as files and directories. Fsync provides data continuity across files and objects, and across heterogeneous data platforms.

Conclusion

Hybrid cloud with Azure allows business owners to augment cloud bursting for scaling chip design workloads. Performance and functional tests with EDA tools and benchmarks on Azure VMs using an Ultra Performance gateway and a 10Gbe ExpressRoute connection from the FlashBlade system in the Equinix cloud-connected data center provide the disaggregation of compute and storage for chip design workloads at scale. The performance test results validated that FlashBlade can scale with respect to capacity and IOPS and throughput.

Data mobility from on-premises FlashBlade to Equinix provides the security, cost efficiency through data reduction, improved data transfer rates for high file count file systems, and automation to configure the data transfer with zero storage touch.

About the Authors

Bikash Roy Choudhury is a technical director at Pure Storage. He is responsible for designing and architecting solutions for HPC/EDA and DevOps workflows that are relevant across industry verticals, including high tech, financial services, gaming, social media, and web-based development organizations that address customer business requirements for on-premises, hybrid, and public cloud environments.

Dr. Ravi Poddar is the Director of HPC Solutions on the FlashBlade team at Pure Storage where he helps develop optimized and efficient solutions for life science, semiconductor, generalized HPC and AI use cases. He has also held Director of Engineering positions at Integrated Device Technology and Transmeta, working in the areas of ASIC and custom design. Ravi received his BSEE with highest honors, MSEE and Ph.D. degrees from the Georgia Institute of Technology. He has over 20 publications and presentations in peer reviewed journals and conferences.

Raymond Tsai is a senior program manager at Microsoft in the Azure Global group specializing in HPC. He has 20+ years of experience in HPC/EDA, AI, and machine learning. His engagements include almost all the major semiconductor partners and end-users worldwide. He has also published several technical articles and has been invited to serve as a lecturer in colleges.

Michael Requa is a principal program manager at Microsoft in the Azure Core Compute group specializing in HPC. His team in Azure qualifies HPC storage technologies and performs end-to-end platform validation for HPC/AI workloads. Before joining Microsoft, Michael was part of two successful startups: BioNanoGenomics (serving as director of bioinformatics), which launched an IPO in 2015, and CycleComputing (serving as a solution architect), which was acquired by Microsoft in 2017.

Giancarlo DiPasquale has over 20 years' experience architecting and delivering enterprise compute and storage solutions. As part of the electronic design automation/semiconductor industry team at Microsoft, Giancarlo is customer-focused and leads the technical effort for the Azure for Silicon initiative, working with semiconductor customers and EDA software vendors to fully realize the possibilities of advanced chip design on Azure cloud.

Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041