PURESTORAGE®
Uncomplicate Data Storage, Forever

NVIDIA Partner

TECHNICAL WHITE PAPER

Accelerating Whole Genome Sequencing with NVIDIA and Pure Storage

Achieve a 33x faster turnaround time for secondary analysis.

Contents

Introduction

Pure Storage® and NVIDIA assessed genome sequencing turnaround times with NVIDIA Clara Parabricks on Pure Storage FlashBlade//S™.

Clara Parabricks is a suite of GPU-accelerated, deep learning genomic tools for secondary analysis with germline, somatic, and RNA applications. FlashBlade® is a consolidated storage platform for file and object workloads, delivering a simplified experience for infrastructure and data management. FlashBlade//S is used by healthcare and life sciences customers to support data-intensive workloads in addition to genomic sequencing analytics, including machine learning and other advanced analytics.

Parabricks software can be run on a variety of NVIDIA GPUs (graphics processing units) and is the only GPU-accelerated computational genomics software suite that includes gold-standard tools like accelerated GATK (Genomics Analysis Toolkit) and DeepVariant for variant calling. Our research compared the performance of Parabricks to a CPU-based (central processing unit) environment.

Our objective is to scale genomic sequencing to support population-level genome sequencing at optimal acceleration with operational efficiency.

We conducted testing in a high-performance computing (HPC) environment on five different next generation sequencing platforms—normal coverage (genome sequencing—NC) samples.

Results

Our test results indicate that GPU-based Clara Parabricks on FlashBlade//S achieves clinical sequencing turnaround times (TAT) that are up to 33x faster than comparable CPU-based alternatives.

FlashBlade//S provided distributed data access to various Clara Parabricks pipelines over high bandwidth and at scale. Improvements in turnaround times were due to the all-flash performance and capacity found in FlashBlade//S, as well as its variable block metadata engine and scale out metadata architecture. Purity//FB, the heart of FlashBlade//S, can handle billions of files and objects and delivers unmatched performance for any workload, whether it's sequential or random, with highly parallel access and ultra low latencies. FlashBlade//S stored the raw data used in the secondary analysis phase and scaled in capacity when the Clara Parabricks pipelines generated a large volume of data.

Why We Conducted This Research

There has been a surge in the volume of genome sample data that needs to be sequenced and analyzed. It's estimated that genomics will generate up to 40 exabytes of data per year by 2025[1]. Comprehensive genome sequencing tools process hundreds of parallel threads simultaneously and on demand. When genome sequencing tools are supported by hundreds of CPUs, research organizations encounter limits related to scale, speed, and simplicity.

1 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4494865/

Multi-core CPUs are mainly designed with shared memory for low latency and are not suitable for parallel processing. Following Moore's law, the packing density of the number of transistors on silicon is running out of momentum. When organizations use hundreds of multi-core CPUs to scale parallel computations sequencing, the compute environment becomes complex and requires more overhead to maintain, including management costs and additional power, cooling, and rack space.

Genome Sequencing with GPUs in HPC Environments

High-performance graphics processing units (GPUs) have enhanced the parallel analyses of multiple genome samples with millions of DNA and RNA molecules. For compute-intensive genomics data, GPU-based sequencing provides a higher degree of parallelism over the traditional CPU. When compared to CPUs, GPU-based architecture provides higher bandwidth with dedicated memory for parallel data access.

Problems in genomics are well-suited for GPUs since they can be broken down into smaller modules with parallel access, increasing efficiency and resulting in a more streamlined infrastructure. Furthermore, the work can be easily scaled across multiple GPUs, reducing compute times.

For customers seeking a converged infrastructure, NVIDIA and Pure Storage offer AIRI//S™, an AI-ready infrastructure that accelerates AI outcomes.

Data Storage and HPC Sequencing Workloads

Data storage is another major component in HPC sequencing workloads. Organizations use data storage in several different ways:

- Temporary storage is used as scratch space for some of the analytics phases for each genome sample. Local SSD-based storage is often the preferred choice for storing the temporary files for long-running tasks.
- Operations storage is mainly used for sharing sample data either between users in the group or across different groups. This type of storage is also used for capacity scaling for genome sample data that can be processed by many GPUs in parallel.
- Archival storage is used for long-term data retention for protection, governance, and cloud extensions. The data can be archived in S3 buckets or blob storage in either public cloud providers or cheap and deep storage on-premises.

During the lifetime of the sequencing process, temporary and operational storage can be configured for many samples in parallel. FlashBlade//S can be used as a standard data platform that removes the time and complexity of moving data between different genome sequence phases. A best practice is to store temporary files on FlashBlade//S when sequencing larger samples, contributing to increased sequencing speeds.

Following secondary analysis, array and host-level data mobility tools are available to move data from FlashBlade//S to other locations for tertiary analysis, including on-premises storage or to the cloud for archiving and extensions to machine learning and analysis.

Methodology

The genome sequencing process consists of primarily three phases: primary, secondary, and tertiary. Our testing focused on the secondary analysis phase.

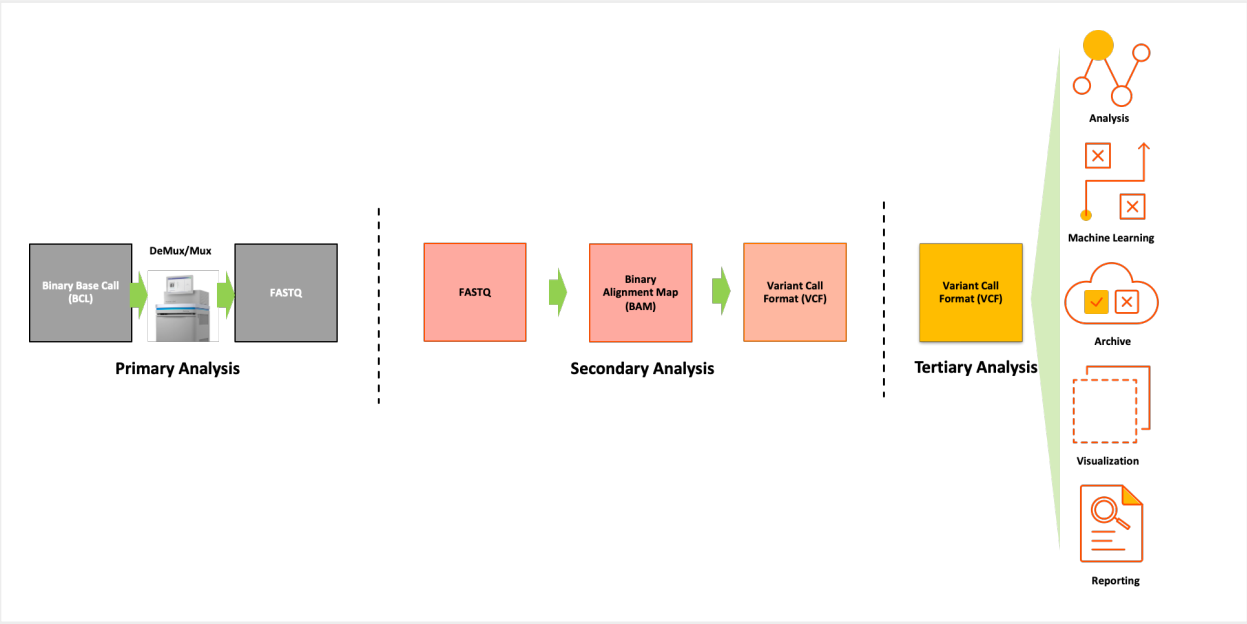

FIGURE 1 Genome sequencing phases—primary, secondary, and tertiary analysis

Three Phases of Genome Sequencing

1. **Primary analysis phase**: This phase includes collecting different raw clinical samples from various handheld input devices and feeding the data into a sequencer. The sequencer takes the binary base call (BCL) files as input and demultiplexes/multiplexes into FASTQ files.
2. **Secondary analysis phase**: This phase uses the FASTQ files from the previous phase and converts them into variant call format (VCF) files.
3. **Tertiary analysis phase**: This phase feeds the VCF files into various machine learning pipelines. These ML pipelines perform complex scientific procedures to analyze, visualize, and understand the results extracted from raw genomics samples. The data is also archived in the cloud for long-term retention.

During our testing, the secondary pipeline read the FASTQ files from FlashBlade//S, sort/align using Burrows-Wheeler aligner (BWA) to binary alignment map (BAM) files that stored sequence data in binary format. The BAM files fed into a GPU-accelerated haplotypecaller for variant calling and written to the FlashBlade//S as variant call format (VCF) files. The secondary analysis phase was automated using various published pipelines in the NVIDIA Clara Parabricks suite.

Clara Parabricks with NVIDIA DGX A100 on FlashBlade//S

We used the genome sequencing samples listed in Table 1 to test performance of the Clara Parabricks germline pipeline running on the GPU-based NVIDIA DGX A100 platform vs. GATK on a CPU-based platform. The NVIDIA DGX A100 includes eight NVIDIA GPUs and two second Gen AMD EPYC processors. We used samples from the Genome in a Bottle project, a publicly available dataset of common genomes for research. Table 1 shows the sample names, as well as the size of each sample in gigabytes (GB) and coverage (how many times the sample was sequenced).

Sample Name		Sample Size	
Coriell ID	**NIST ID**	**GB**	**Coverage**
NA12878	HG001	65	41.8x
NA24385	HG002	72	32.1x
NA24149	HG003	110	42.3x
NA24143	HG004	36	16.6x
NA24631	HG005	39	18.7x

TABLE 1 Genome sequencing sample test data

Test Environment and Best Practices

1. The DGX A100 was configured to mount filesystems on the FlashBlade//S for each of the samples over NFSv3.
2. Clara Parabricks 4.0.0.1 was downloaded and configured on DGX server.
3. The FASTQ files for every sample were used as input files to the pipeline.
4. The workload was scaled from 2-4-8 GPUs for every sample.
5. The temporary storage for scratch was configured on FlashBlade//S over NFSv3.
6. The hardware used for CPU-based validation is well above the GATK-recommended configurations.

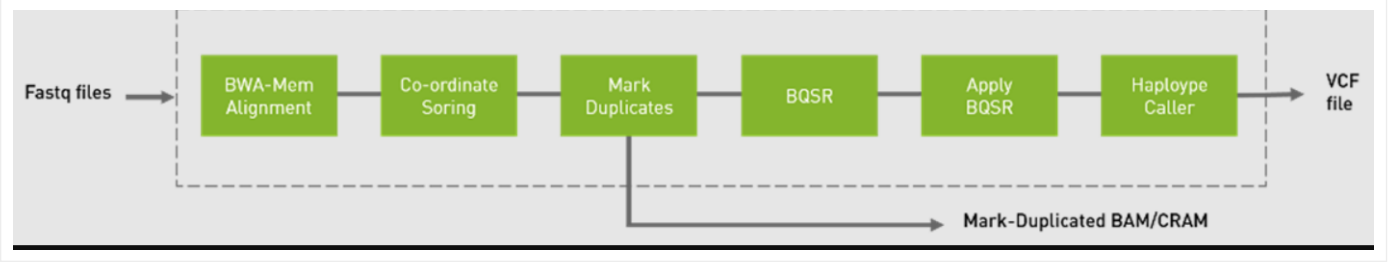

FIGURE 2 Clara Parabricks Germline Pipeline

During the secondary analysis phase, the FASTQ files were read from the FlashBlade//S. The workload's read and write operations were highly sequential. We recommend using the largest available IO block size for effective readahead capability from the DGX Ubuntu server. By default, the DGX Ubuntu server runs on v20.04, which is limited with a readahead size of 128k.

The mount options were set to the default of 512k rsize/wsize for all the NFS mount paths for the five genome samples. (We recommend setting the readahead IO block size to 512k for improved performance.) The readahead value has to be set to 7680. This value is derived from scaling 15 * 512k (rsize/wsize) = 7680. A `99-nfs.rules` file `under /etc/etc/udev/rules.d/` is created to maintain persistence across reboots.

```
root@sn1-dgx-a100-a03-07:/mnt/NA12878# cd /etc/udev/rules.d/
root@sn1-dgx-a100-a03-07:/etc/udev/rules.d# vi 99-nfs.rules
SUBSYSTEM=="bdi", ACTION=="add", PROGRAM="/bin/awk -v bdi=$kernel 'BEGIN{ret=1} {if ($4 == bdi) {ret=0}}
END{exit ret}' /proc/fs/nfsfs/volumes", ATTR{read_ahead_kb}="7680"
```

We used the following command to apply the new rule:

```
root@sn1-dgx-a100-a03-07:/etc/udev/rules.d# udevadm control --reload
root@sn1-dgx-a100-a03-07:/etc/udev/rules.d#
```

For benchmarking, we used the following two flags. The --run-partition flag split the workload into multiple partitions and ran each partition with its own process on two of the GPUs. The --no-alt-contigs flag eliminated any processing of alternate contig sequences, which are more complex ways to represent parts of the genome.

```
--run-partition
--no-alt-contigs
```

Test Results and Observations

The following graphs illustrate the completion time for each sample while scaling from 2-4-8 GPUs and compared with the CPU performance.

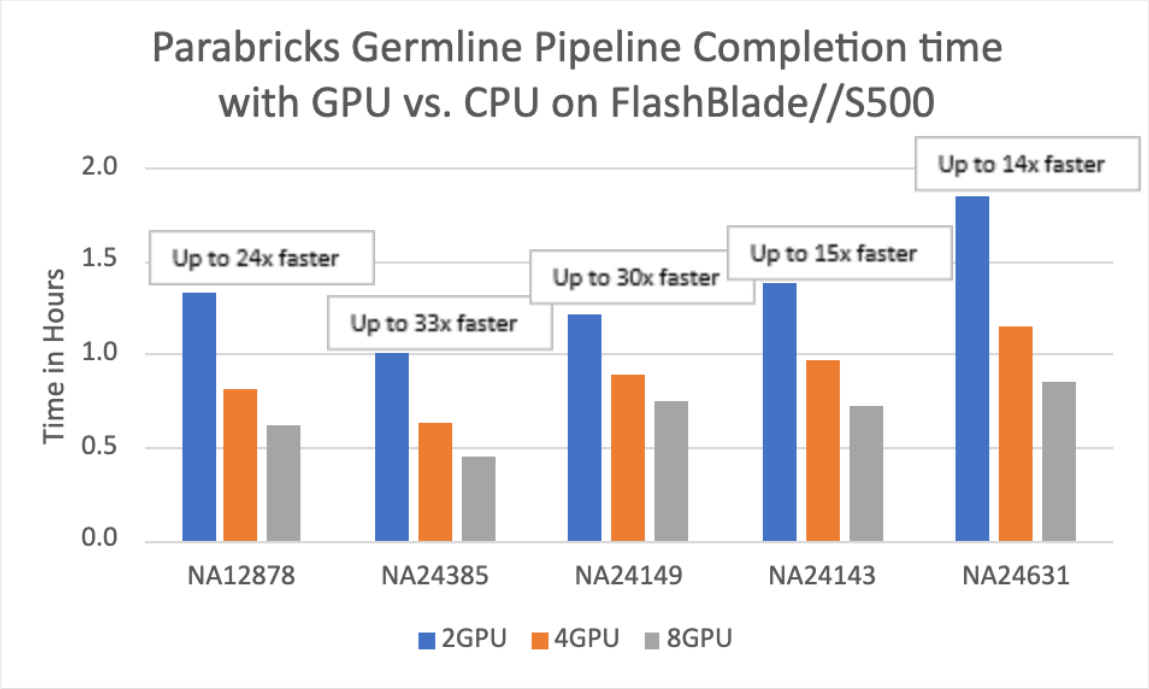

FIGURE 3 Completion for Clara Parabricks Germline Pipeline on DGX A100 with FlashBlade//S

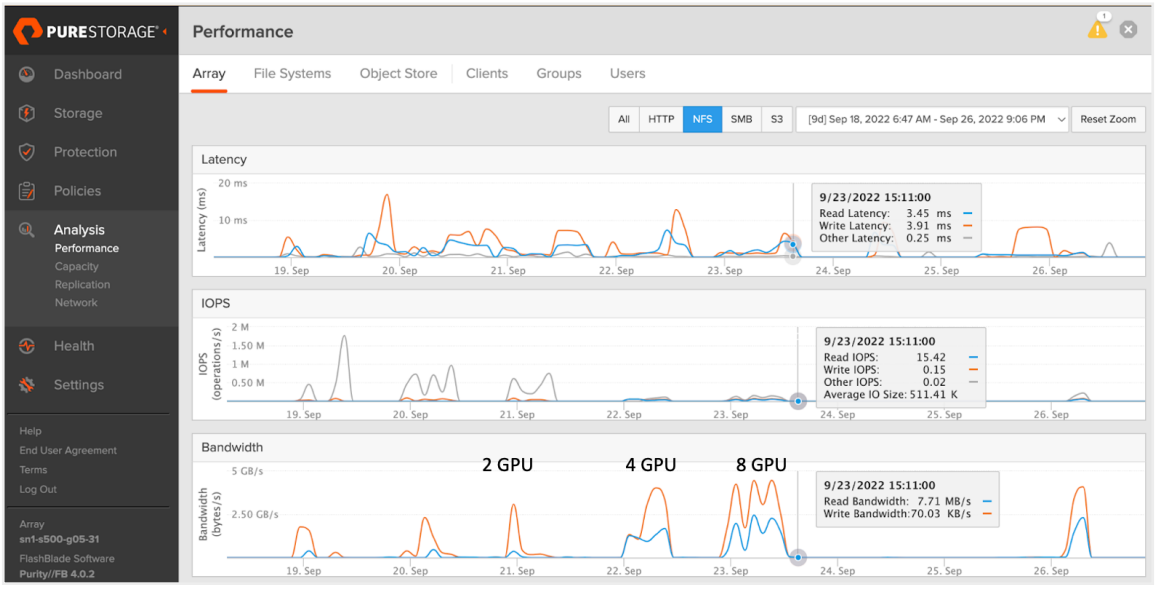

FIGURE 4 ~Linear scalability with FlashBlade//S for genome samples

Main Observations

We observed the following.

1. Each of the genome samples had a faster completion time compared to the CPU-based platform during the secondary analysis phase. There was further time reduction as the GPUs scaled from 2-4-8. The 8-GPU recorded faster completion time for all the samples.
2. During the Clara Parabricks germline pipeline runs, the FlashBlade//S demonstrated high throughput (aggregate read/write throughput recorded at 7GB/sec) with tolerable read and write latencies with 512k IO sizes. This clearly indicated how FlashBlade//S linearly scales in performance with GPUs, when genome sequencing is conducted in an HPC environment.
3. The scratch space used by the BWA-mem and HaplotypeCaller for its long-running tasks were mainly single-threaded operations for the MarkDuplicates. The FlashBlade//S handled the single thread operations at high speed. There was no difference in the overall completion time with the temporary space configured on local storage compared to the FlashBlade//S.
4. While focusing on a single genome sample (NA24385), the completion time scaled linearly with GPUs compared to CPUs. This genome sample had a coverage depth of 32.1x and could run 595 genomes/year in a CPU cluster compared to 19466 genomes/year on a DGX A100 with the all-flash FlashBlade//S. There was a 33x improvement in a GPU-based HPC-enabled Clara Parabricks environment compared to a CPU-based cluster.

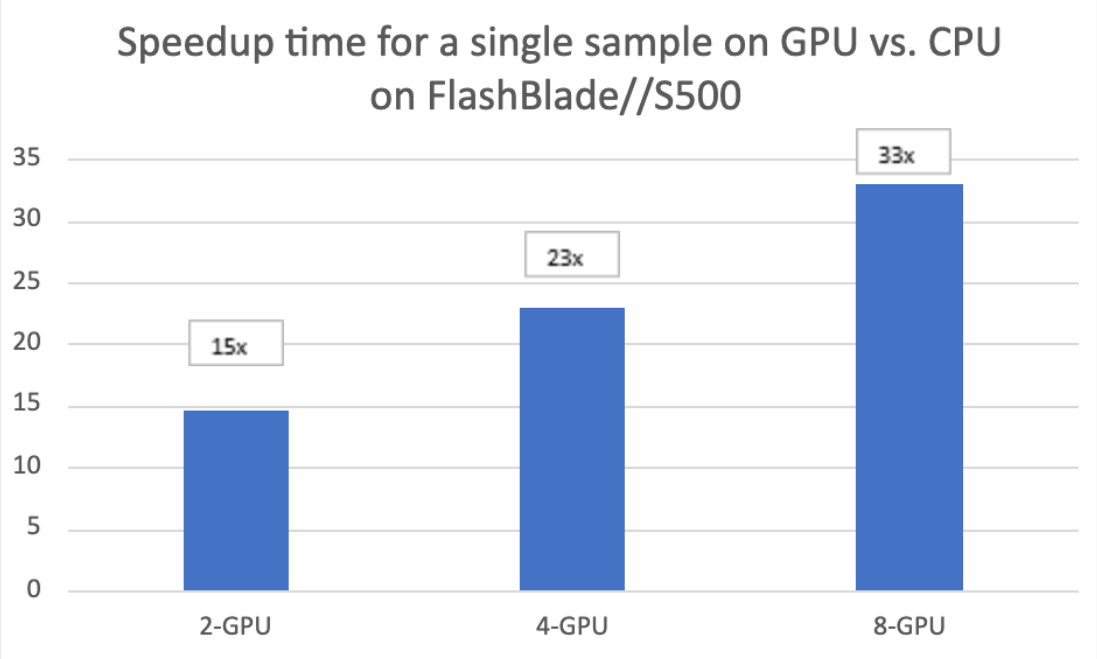

FIGURE 5 ~Speedup time for a single sample on GPU vs CPU

Accuracy with SNP and INDEL

The following graph shows the accuracy of the Clara Parabricks variant calling using the GATK variants as the ground truth. This was measured across two types of variants: SNPs (single nucleotide polymorphisms or point mutations) and indels (insertions/deletions). The metrics are given in terms of precision and recall.

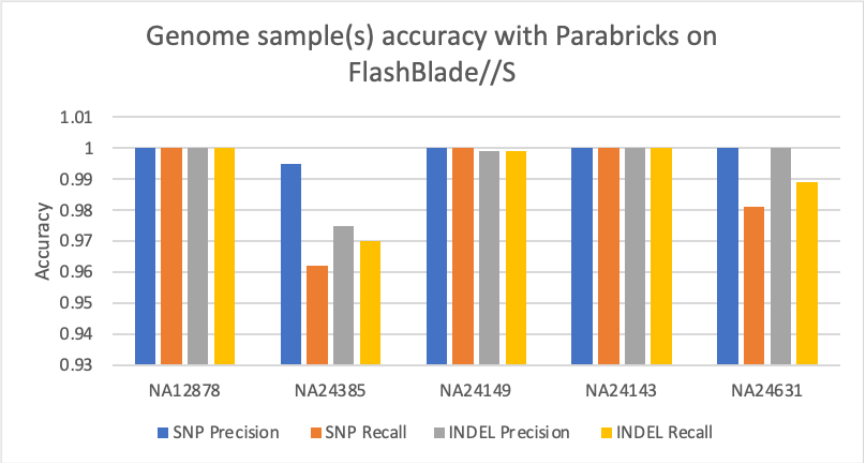

FIGURE 6 Genome sample accuracy

GATK Observations

Clara Parabricks resulted in nearly identical variants as GATK, but in less time by orders of magnitude. This bar graph shows that all of the samples reached at least 96% accuracy for precision and recall, with most showing 100% agreement with GATK. When the GPU-based acceleration of the genome sequence accuracy is complemented by a distributed file system and an all-flash data platform like FlashBlade//S, faster results are achieved.

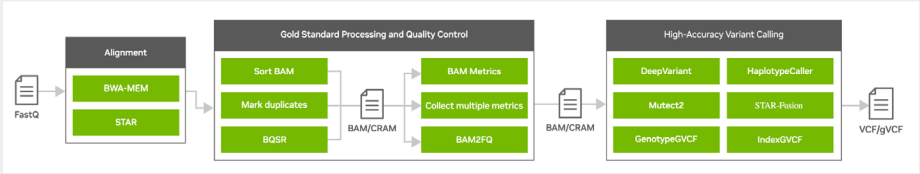

FIGURE 7 NVIDIA Clara Parabricks GPU-accelerated tools for DNA and RNA applications

Conclusion

The GPU-accelerated WGS process with Clara Parabricks on DGX A100 and FlashBlade//S is rapidly changing the HPC landscape, supporting faster and more efficient genome sequencing.

The test results documented in this paper show that GPU-optimized Clara Parabricks on FlashBlade//S demonstrates a high degree of parallelism in performing complex computations at massive speed, significantly reducing the sequencing time for each sample. The FlashBlade//S stores raw data used in the secondary analysis phase and scales in capacity when the Clara Parabricks pipelines generate a large volume of data. The FASTQ files can be stored on FlashBlade//S for the long term and be used by different clinical labs. This approach provides greater access to shared data, rapidly creates results for patients, and reduces costs from WGS clinical tests.

PS2401-01-en 03/23