WHITE PAPER

Broad-Intel Genomics Stack
High-Performance Computing

Accelerate Genomics Research with the Broad-Intel Genomics Stack*

Learn how the Broad-Intel Genomics Stack* (BIGstack*), running on the latest-generation Intel® Xeon® processors and other Intel® technologies, can improve latency and throughput—powering new insights quickly

intel select solution

The highly-optimized BIGstack* with Intel® Xeon® Scalable processors and Intel® SSDs delivers an impressive throughput of up to 5 whole genomes per day per node, while using Intel® FPGA technology to accelerate the analysis by up to 2.2x over prior-generation Intel® Xeon® processors and up to 1.8x over hard drives.

Authors

Patrick Foley
Abirami Prabhakaran
Karthik Gururaj
Mishali Naik
Shiva Gopalan
Aleksandr Shargorodskiy
Ernesto Brau

Executive Overview

Genomics is revolutionizing our understanding of human biology and contributing to the growth of precision medicine. But the sheer amount of data associated with genomics research has historically limited the pace at which new insights are obtained. It took 13 years and USD 3 billion to sequence the first human genome; as recently as 2012, only 69 whole human genomes had been sequenced. Yet, recently researchers at University of Toronto launched a massive project to sequence the genomes of 10,000 people per year.[1]

Intel and the Broad Institute of MIT and Harvard are at the forefront of the effort to accelerate genomics analysis and the benefits it can produce. Together, Intel and Broad have introduced an integrated hardware and software solution to run Broad's popular Genome Analysis Toolkit* (GATK*) faster, at unprecedented scale, and with easier deployment. It used to take six weeks to generate a database from 2,300 genomes. Now, using the Broad-Intel Genomics Stack* (BIGstack*), a database containing 5x more information can be generated in only two weeks.[2]

BIGstack is a game-changing, end-to-end integrated hardware and software package. With common, validated reference designs that use the latest-generation Intel® Xeon® Scalable processors, Intel® Arria® 10 Field Programmable Gate Array (FPGA) PCIe* cards, Intel® Omni-Path Architecture (Intel® OPA), and Intel® 3D NAND Solid State Drives (Intel® SSDs), BIGstack can help ease the complexity of running the genomics analysis pipeline (specifically, Broad Institute's production-worthy Best Practices workflows) while dramatically speeding up the analysis process.

This paper demonstrates how a BIGstack-based platform that uses the latest Intel Xeon Scalable processors and Intel 3D NAND SSDs achieves a throughput of up to 5 whole genomes and more than 100 whole exomes per day per node. Intel® FPGA technology further speeds up the individual sample analysis by up to 2.2x for whole genomes at a lower memory cost compared to prior-generation Intel® Xeon® processor for Broad's GATK Best Practices. Information is provided about tools, technologies, optimizations, and methodology, as well as details about latency, throughput, and utilization of CPU, memory, and disk resources.

Table of Contents

Introduction

The Broad-Intel Genomics Stack* (BIGstack*) is an end-to-end, optimized hardware and software solution for analyzing genomic data. It provides a way to run pre-packaged, optimized workflows, including the Genome Analysis Toolkit* (GATK*) Best Practices workflow from the Broad Institute. These complex workflows target specific scientific questions of interest to the genomics community, and support data analysis using commonly available analysis tools.

BIGstack includes two components developed by Intel:

- **Genomics Kernel Library*** (GKL*) accelerates commonly used, compute-intensive genomics kernels on Intel® architecture.[4]
- **GenomicsDB*** provides a scalable and efficient means to store genomic variants.[5]

In addition to GATK[6], BIGstack also supports other open source libraries of genomic analysis tools, such as Picard*[7], BWA*[8], and Samtools*[9]. These tools perform a wide variety of tasks, from sorting and fixing tags to generating recalibration models. Users specify the files to be analyzed, what tools they want to use, and the order in which the execution engine (Cromwell*)[10] performs the tasks using Workflow Description Language (WDL)[11] files. WDL is a standard developed by the Broad Institute specifically for genomic analysis and is designed to be easy to use. Intel is providing Broad-generated WDL workflows so users can quickly and easily deploy Broad's high-quality pipelines. With WDL and Cromwell, users can also implement their unique pipelines using whatever tools and steps best meet the user's requirements. Intel will continue to expand the WDL offering, providing additional pipelines and flexibility in the future.

The following sections describe that hardware and software used in this paper, followed by the Results section, which provides a thorough analysis of the performance across a number of different hardware configurations. The results clearly indicate that the latest-generation Intel® Xeon® Scalable processor, combined with Intel® Arria® 10 FPGAs (field programmable field arrays) and Intel® 3D NAND Solid State Drives (Intel® SSDs) featuring Non-Volatile Memory Express* (NVMe*), provides the lowest latency and best throughput for Broad's Best Practices workflow, compared to prior generations of processor and hard disk drives (HDDs).

Hardware

Table 1 provides configuration information for the nodes used in the benchmarking. The first two columns represent the latest-generation Intel Xeon Scalable processor platform across all system components. The leftmost column utilizes an Intel Arria 10 FPGA PCIe* card, and includes one fewer SSD due to space constraints as compared to the middle column. The rightmost column is configured using the fastest prior-generation Intel® Xeon® processor. The latest-generation Intel 3D NAND SSD-based storage configuration 1 is used for all SSD experiments, and the hard drive configuration described as storage configuration 2 is used for all hard drive experiments.

Workflows and Data

The GATK Best Practices workflow is available through the Broad Institute. For the experiments in this paper, we used two versions of the Single-Sample Germline Variant Calling workflow[12]:

- A purely GATK v3.8-based workflow for the Whole Exome Sequencing (WES) data
- A hybrid workflow that uses both GATK v.3.8 and GATK v4.0 for the Whole Genome Sequencing (WGS) data

Table 1. Benchmarking Configurations

	Latest-Generation Platform Configuration with Intel® Arria® 10 FPGA PCIe* Card	Latest-Generation Platform Configuration	Prior-Generation Platform Configuration
Number of Nodes	1	1	1
Chassis	R2308WFTZS	R2308WFTZS	R2308WTTYS
System Board	S2600WFT	S2600WFT	S2600WTTR
Processor	2 x Intel® Xeon® Platinum 8180 Processor with 28 cores each (56 total)	2 x Intel Xeon Platinum 8180 Processor with 28 cores each (56 total)	2 x Intel® Xeon® Processor E5-2699 v4 with 22 cores each (44 total)
FPGA	1 x Intel Arria 10 FPGA PCIe Card	NA	NA
Memory	8/16 x 32 GB 2666 REG ECC (Total 256/512 GB)	16 x 32 GB 2666 REG ECC (Total 512 GB)	16 x 32 GB 2400 REG ECC (Total 512 GB)
Storage Configuration 1	7 x Intel® SSD Data Center P4600 Series 2 TB HHHL PCIe* (Total 14 TB)	8 x Intel SSD Data Center P4600 Series 4 TB HHHL PCIe (Total 32 TB)	8 x Intel SSD Data Center P4600 Series 4 TB HHHL PCIe (Total 32 TB)
Storage Configuration 2	NA	8 x Western Digital* 6 TB SAS HDD 3.5" (Total 48 TB)	NA

ECC – Error-correcting code; **FPGA** – field programmable gate array; **HDD** – hard disk drive; **HHHL** – half height, half length; **SAS** – serial attached SCSI; **SSD** – Solid State Drive

Both versions of the Single-Sample Germline Variant Calling workflow effectively stream the per-sample input data to pre-processing and variant calling segments of the workflow to produce a single gVCF file, and are implemented using WDL. Appendix B and Appendix C describe the flow of these two workflows in detail.

The unmapped WGS and WES BAM datasets used for both workflows are listed in Table A1 and Table A2 in Appendix A. The reference and resource files used for this workflow are the HG38 bundle:

- Inputs:
  - HG38 reference genome
  - VCF files: DBSNP, Mills Indels, HapMap, Omni, 1000G Phase SNPs
  - WGS and WES interval files
- Outputs:
  - Processed VCF files (produced by the MergeVCF step of the Single-Sample Germline Variant Calling workflow)

Tools and Technologies

Table 2 lists the genomic tools and technologies used by the Single-Sample Germline Variant Calling workflow.

Appendix E provides more information on the tools and technologies used by the Single-Sample Germline Variant Calling workflow.

Experimental Methodology

Figure 1 illustrates the environment setup used for the experiments in this paper. The workflows are submitted to the Cromwell (v28) execution engine, which uses the HTCondor backend (v8.6.1)[13] to allocate workflow tasks among available compute resources.

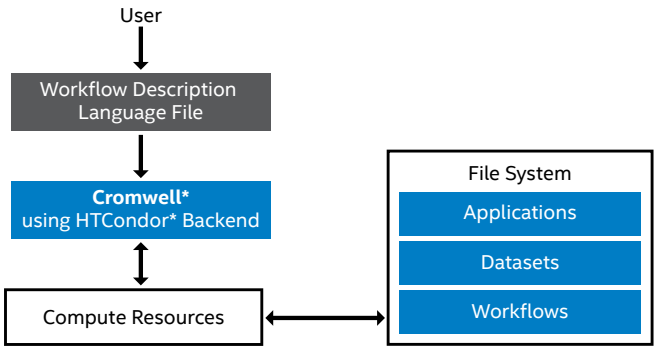


Figure 1. Experimental setup.

The workflows are configured and optimized for lower latency and higher throughput. For the latency experiments, the tasks that benefit from parallelism are optimized for speed. For the throughput experiments, the tasks are given fewer resources, which enables simultaneous execution of more tasks on the node.

The benchmarking described in this paper (see the Results section) focuses on the following metrics for the Single-Sample Germline Variant Calling in Broad's Best Practices workflow:

- Latency:
  - Single workflow latency comparison benchmark between Intel SSDs with NVMe and HDDs on WES and WGS.
  - Single workflow latency comparison benchmark between the platform based on the prior-generation Intel® Xeon® processor E5-2699 v4 and a platform based on the new generation Intel® Xeon® Platinum 8180 processor with Intel® FPGA technology on WES and WGS.
- Throughput:
  - Comparison to estimate the number of whole genomes and whole exomes processed in a day using Intel SSDs and HDDs.
  - Comparison to estimate the number of whole genomes and whole exomes processed in a day using the older platform based on the Intel Xeon processor E5-2699 v4 and the platform based on the new generation Intel Xeon Platinum 8180 processor.

Results for CPU, memory, and disk utilization are also provided for these experiments.

Results

In this section, we discuss the performance of the Broad Institute's GATK Best Practices workflow on the three hardware platforms described in the Hardware section. Results are provided for latency and throughput comparison across CPU generations, and across different storage configurations. CPU utilization, memory utilization, and read/write performance graphs and in-depth analysis are provided for the Hybrid workflow running the WGS dataset.

Table 2. Tools and Technologies Versions

Tool	Version for Whole Exome Sequencing (WES) GATK 3.8 Workflow	Version for Whole Genome Sequencing (WGS) Hybrid Workflow
BWA*	0.7.15-r1140	0.7.15-r1140
Picard*	2.8.3-SNAPSHOT	2.12.0-SNAPSHOT
Samtools*	1.3.1, using htslib 1.3.1	1.3.1, using htslib 1.3.1
GATK*	3.8-1	3.8-1 and 4.0 beta.5
Python*	2.7	2.7
Genomic Kernel Library*	0.6.0	0.6.0
Java*	OpenJDK build 1.8.0 151-b12	OpenJDK build 1.8.0 151-b12

LATENCY

In this section we focus on latency experiments for a single WGS and WES workflow. To maximize the CPU utilization, we extensively profiled and tuned each of the stages in the workflow, and set the CPU and memory requirements accordingly. SamToFastqAndBwaMemandMba and SortSampleBam stages consisting of BWA-MEM and Samtools Sort applications scale well with the number of threads; hence, these workflow stages are allocated all the available cores on the system (that is, 56 and 44 threads/cores for the Intel Xeon Platinum 8180 processor and Intel Xeon processor E5-2699 v4 experiments, respectively). The BWA-MEM application is allotted 54 threads for the Intel Xeon Platinum 8180 processor and 42 threads for the Intel Xeon processor E5-2699 v4 to accommodate the other applications running simultaneously in the SamToFastqAndBwaMemAndMba stage. Because all other stages are bound by either memory or disk, we only allocate to them one thread/core.

Furthermore, to enable a fair comparison between the hardware capabilities of SSDs and HDDs, we made several HDD-specific workflow and system configuration optimizations that were not implemented on the configurations that utilized SSDs. These HDD-specific optimizations include disabling swap space for the workflow applications and increasing the total memory to SortSampleBam as well as the memory to each thread for the SortSampleBam.

For the latency analysis, we focus on WGS workflows to capture bottlenecks. WES data is much smaller than WGS data (5.9 GB versus 64 GB), so system bottlenecks get masked. For example, for WES workflows, the data can easily fit in the 512 GB of memory on the system, so the CPU is only blocked by disk for a short time at the beginning and end of each workflow task. Some of the workarounds that had to be employed for WGS workflows, when the allotted memory size was exceeded (which led to job resubmission), did not need to be employed for WES data. In addition, the WGS workflow uses the newer Hybrid workflow, which spends a greater percentage of time in stages optimized for the latest Intel architecture.

With the development of BIGstack, Intel has made significant optimizations to the GKL to accelerate the PairHMM algorithm for Intel® Advanced Vector Extensions (Intel® AVX), Intel® Advanced Vector Extensions 512 (Intel® AVX-512), and the Intel Arria 10 FPGA. The HaplotypeCaller stage of the GATK Best Practices workflow makes extensive use of the PairHMM algorithm, as evidenced by the results shown in Figure 2. Using WGS data, we see a 2.2x speedup overall in the HaplotypeCaller stage when using the Intel Arria 10 FPGA compared to prior-generation Intel AVX technology. For this reason, our experiments use the Intel Xeon Platinum 8180 processor with the Intel Arria 10 FPGA hardware configuration in the following latency experiments.

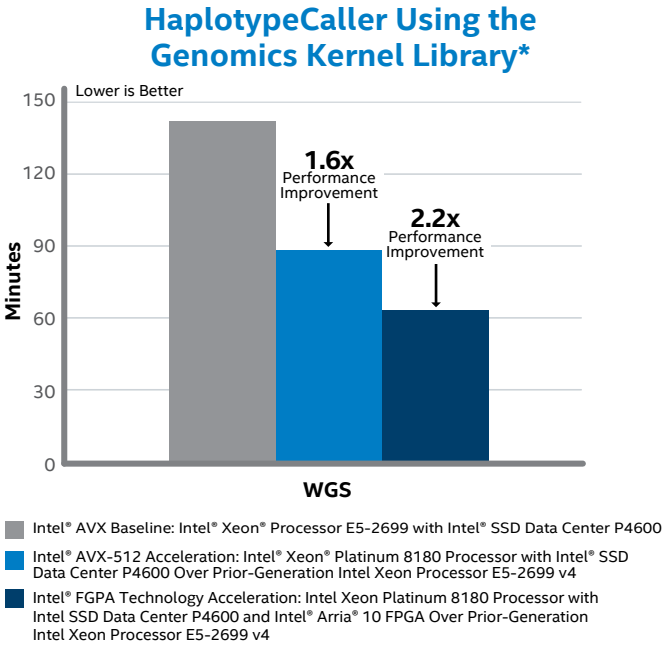


Figure 2. Execution time comparison for HaplotypeCaller using the Genomics Kernel Library*.

GATK* Best Practices Workflow - 3.3x Faster

The Genome Analysis Toolkit* (GATK*) Best Practices from the Broad Institute focuses on standardizing the methods by which sequencing data is analyzed. Intel has worked with the Broad Institute on accelerating the commonly used, compute-intensive genomics kernels on Intel® architecture, as well as optimizing and benchmarking the Best Practices workflows on the latest-generation Intel® reference hardware platform.

As shown in Figure 4, with the latest reference architecture based on the Intel® Xeon® Platinum 8180 processor, Intel® Arria® 10 Field Programmable Gate Array (FPGA), and Intel® Solid State Drives (Intel® SSDs), it takes approximately 10.8 hours to process a Whole Genome Sequence (WGS) at 30x coverage using the HG38 reference genome on the latest Hybrid GATK Best Practices workflow for germline variant detection. Comparing this against the white paper Intel published in 2016[14], which showcased the "then" GATK Best Practices workflow for Whole Genome Sequence (WGS) that took 36.12 hours, we see a compelling 3.3x speedup on the latest available Intel reference hardware.

Figure 3 shows the two stages of the Hybrid workflow that benefit the most from Intel SSDs compared to HDDs. On the platform using Intel Xeon Platinum 8180 processor with SSDs, Samtools Sort and GatherBQSRReports stages of the workflow provide up to 1.8x speedup over HDDs. Overall, SSDs provide up to a 1.2x speedup over HDDs for this entire Hybrid workflow.

Figure 4 compares the execution time in hours taken to complete the end-to-end processing of a single WGS and WES workflow using the latest-generation Intel Xeon Platinum 8180 processor with Intel FPGA technology and the prior-generation Intel Xeon processor E5-2699 v4. The newer processor provides a 1.3x and 1.2x speedup over the prior-generation processor for WGS and WES, respectively.

Furthermore, comparing the blue bars in Figure 4, we observe that the Intel Xeon Scalable processor-based hardware platform, together with Intel 3D NAND SSDs and Intel FPGA technology configured with 256 GB of memory, delivers very close performance compared to the same hardware platform configured with 512 GB of memory, thus providing a significant opportunity to lower total cost of ownership (TCO).

THROUGHPUT

In this section, we discuss how using the latest-generation Intel Xeon Scalable processor and Intel SSDs with NVMe can increase throughput compared to the prior-generation processor and HDDs. We define the throughput metric as the number of processed samples produced per day per node.

To maximize the number of samples processed per day per node, we adjusted the load to achieve maximum CPU utilization, while taking into account the stages of the workflow that are bound by memory or disk space. We then extensively tuned each of the stages in the workflow by setting the CPU and memory requirements based on the latency experiments for each of these individual workflows. For example, while all the available threads/cores on the system are allocated to the SamToFastqAndBwaMemandMba and SortSampleBam stages in the latency experiments, they are set to request only 2 threads/cores for throughput experiments to enable as many tasks as possible to execute in parallel at any given time. Also, the memory-per-thread for the Samtools Sort stage is tuned for SSDs and HDDs based on the analysis observed in the latency experiments.

Collaborating for Success

The Intel-Broad Center for Genomic Data Engineering is a five-year, USD 25 million collaboration announced in November 2016.[15] During this effort, researchers and software engineers at the new Intel-Broad Center for Genomic Data Engineering will build, optimize, and widely share new tools and infrastructure that will help scientists integrate and process genomic data.

Learn more at **Intel-Broad Center for Genomic Data Engineering.**

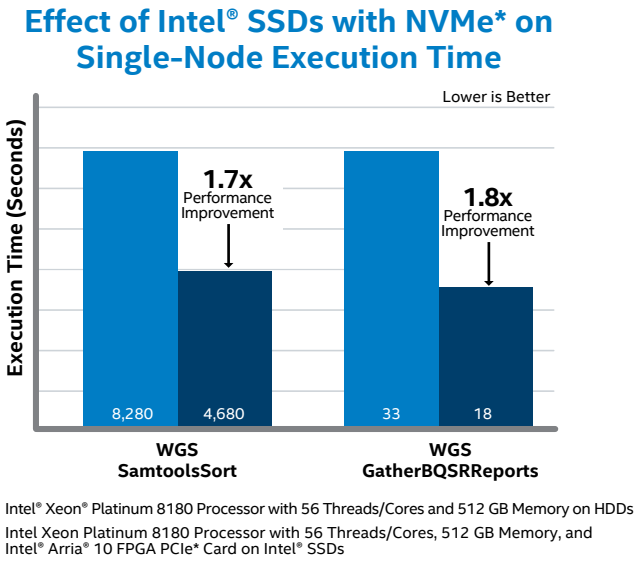


Figure 3. Effect of Intel® Solid State Drives with NVMe* on execution time (platform based on the Intel® Xeon® Platinum 8180 processor).

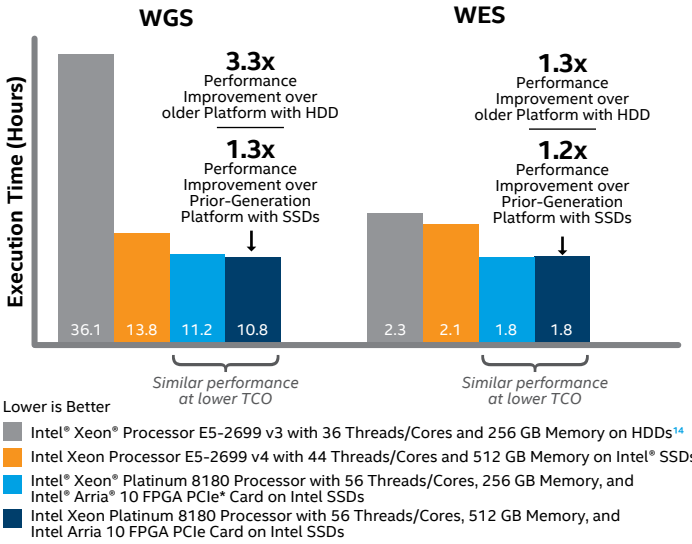


Figure 4. Effect of latest-generation processor on execution time.

Figure 5 compares the throughput achieved using prior-generation and latest-generation Intel Xeon processors. The latest-generation processor provides 1.4x higher throughput over the prior-generation processors for both WGS and WES.

Figure 6 compares the throughput that can be sustained on a platform using the Intel Xeon Platinum 8180 processor equipped with SSDs, compared to HDDs. For the WES workflow, the SSDs deliver close to 1.1x more samples per day compared to HDDs, while for the WGS workflow the SSDs deliver close to 1.2x more samples per day compared to HDDs. The high-performance SSDs help alleviate the I/O contention when multiple samples are contending for resources, thus providing significantly higher throughput than the HDDs.

SYSTEM UTILIZATION FOR LATENCY EXPERIMENTS

The following sections discuss our observations about CPU and memory utilization and read/write performance during our latency experiments.

CPU Utilization

Figure 7 demonstrates the average CPU utilization when executing a single WGS workflow on the different reference architectures described in the Hardware section. The vertical axis of the graph displays the percentage of the CPU used across all processor cores; the horizontal axis represents the number of hours since the workflow batch was submitted. It took a total of 10.85 hours to run a single WGS on an Intel Arria 10 FPGA card with SSDs, and a total of 12.60 hours for the same processor-based platform with HDDs, and 13.82 hours on the Intel Xeon processor E5-2699 v4 platform with SSDs.

In the WGS Hybrid workflow, SamToFastqAndBwaMemAndMba exploits data-parallelism by processing unaligned BAM files by read group. Because there are 24 read groups, and all available processor cores are allocated to this step, we observe 24 peaks and valleys in the first step corresponding to each of the 24 shards. The valleys observed in the light blue line are more dramatic due to longer read and write times required for the HDD. The trend line for the Intel Xeon processor E5-2699 v4 configuration (green line) demonstrates the increased runtime in the SamToFastqAndBwaMemAndMba stage resulting from only allocating 42 threads to BWA-MEM instead of 54 for the other tested configurations. Proceeding to the MarkDuplicates task, we observe sustained CPU utilization of about 20 percent for all tested configurations. Around 7 hours into workflow execution, there is a spike in HDD CPU utilization, and a corresponding spike in memory utilization, which is due to differences in how memory is allocated to the Samtools Sort application. The final steps in the workflow of the SSD and HDD experiments differ primarily in their phase and also slight differences in the duration of these steps. This is primarily due to additional time taken to read and write to disk.

Overall, we observe a clear trend of the Intel Xeon Platinum 8180 processor-based platform configuration with 56 threads/cores and Intel Arria 10 FPGA PCIe card with Intel SSDs (dark blue line) resulting in best latency performance, whereas removing the FPGA and SSDs results in lower performance.

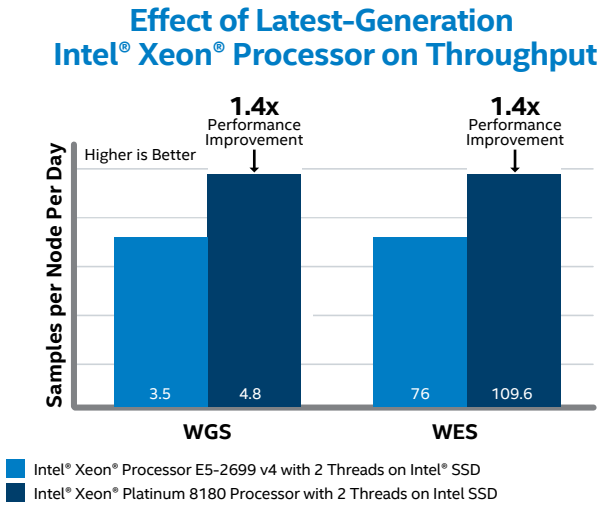


Figure 5. Effect of latest-generation Intel® Xeon® processor on throughput.

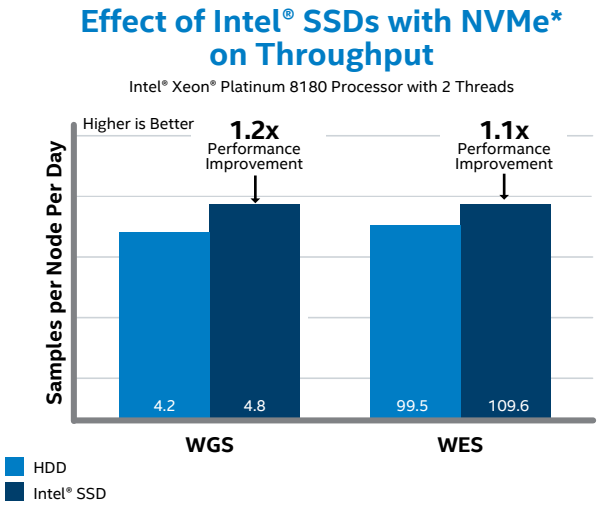


Figure 6. Effect of Intel® SSDs with NVMe* on throughput (platform based on the Intel® Xeon® Platinum 8180 processor).

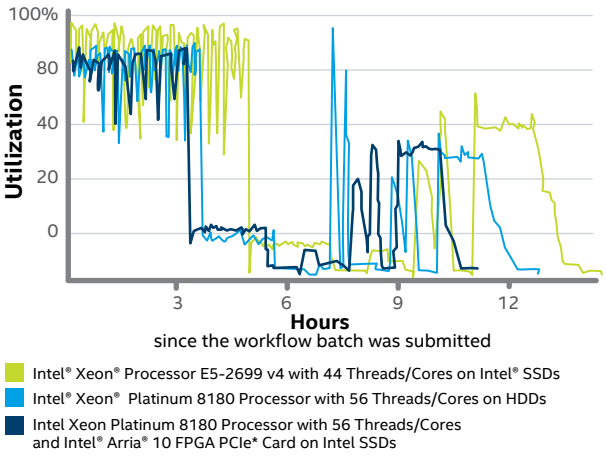


Figure 7. Single Whole Genome Sequence (WGS) latency CPU utilization comparison.

Both configurations utilizing the latest-generation Intel Xeon Platinum 8180 processor result in a double-digit performance increase over the fastest prior-generation Intel Xeon processor.

Memory Utilization

In Figure 8, we provide a comparison of memory utilization across the Intel Xeon Platinum 8180 processor with Intel Arria 10 FPGA and SSDs (dark blue line), the same processor with HDDs (blue line), and the prior-generation Intel Xeon processor E5-2699 v4 with SSDs (orange line). The vertical axis shows the node's memory utilization up to 256 GB, out of a total 512 GB available on each of the tested configurations. The memory utilization for the Intel Xeon Platinum 8180 processor with HDDs (blue line) is slightly more than the same processor with SSDs (dark blue line) because there was no swap space made available to the workflow tasks in the HDD experiments in order to avoid frequent, costly writes to disk; therefore, the red line should be viewed as total amount of memory that the tasks require. The Intel Xeon Platinum 8180 processor with SSDs (dark blue line) effectively uses page swapping to SSD, which results in lower observed memory utilization. The notable exception to this trend is the Samtools Sort stage (between hours 7-8), which allocates significantly more memory per thread when using HDD versus SSD. As mentioned earlier, this correlates strongly with a spike in CPU utilization for the system configured with HDDs.

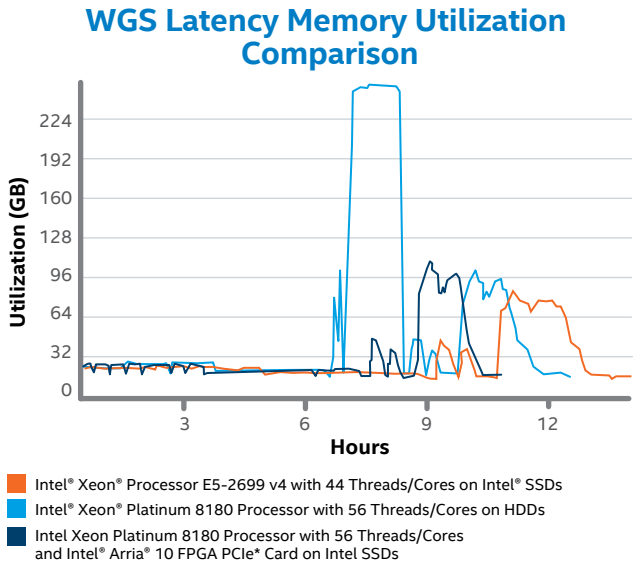


Figure 8. Single Whole Genome Sequence (WGS) latency memory utilization comparison.

Read/Write Performance

The left side of Figure 9 demonstrates the average read and write operations when executing a single WGS workflow on a system using the Intel Xeon Platinum 8180 processor with Intel Arria 10 FPGA PCIe Card and SSDs versus the Intel Xeon processor E5-2699 v4 with SSDs. Analogously, the right side demonstrates the average read and write operations when executing a single WGS workflow on the Intel Xeon Platinum 8180 processor using HDDs. These figures are separated so that is easy to identify subtle differences between workflow stages that are obfuscated when the graphs are combined because of differences in SSD/HDD IOPS scale.

The left side of Figure 9 shows nearly constant writes to the SSD volume in the SamToFastqAndBwaMemAndMba stage for both generations of processors. Whereas, in the right side we see staggered writes to the HDD in this stage. We also observe that the total number of write operations is higher in this stage with SSDs, which can be attributed to the higher bandwidth of SSDs. Because of the way these experiments were configured with swapiness enabled for SSDs and disabled for HDDs, we can easily determine in later workflow steps where the read and write operations are making use of swap space versus when task inputs and outputs are read to and from disk.

For example, we see a major spike between the 6-7 hour mark on the right side of Figure 9 that corresponds to temporary files written to disk in the SortSampleBam stage, as well as a write peak near the end of the graph that aligns with the conclusion of multiple HaplotypeCaller shards. As mentioned in the Memory Utilization section, overall we see much more read and write activity with SSDs throughout the graph because pages are swapped in and out of memory onto the SSD, but there are only a few minor bumps on the right side of Figure 9.

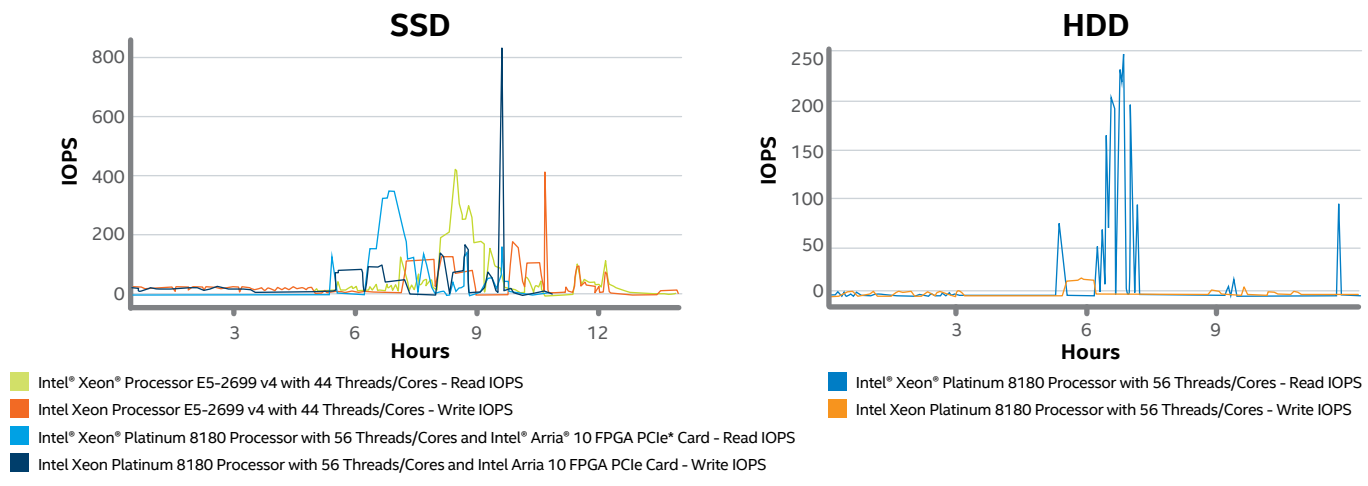


Figure 9. Single Whole Genome Sequence (WGS) latency read/write SSD (left) and HDD (right) IOPS comparison.

SYSTEM UTILIZATION FOR THROUGHPUT EXPERIMENTS

The following sections discuss our observations about CPU and memory utilization and read/write performance during our throughput experiments

CPU Utilization

Figure 10 demonstrates the average CPU utilization with a load of 50 WGS workflows across the different reference architectures described in the Hardware section. The vertical axis displays the percentage of the CPU used across all processor cores; the horizontal axis represents the number of days since the workflow batch was submitted. It took a total of 10.35 days to process the 50 WGS workflows on the Intel Xeon Platinum 8180 processor platform with SSDs; 11.87 days for the Intel Xeon Platinum 8180 processor platform with HDDs; and 14 days for the Intel Xeon processor E5-2699 v4 platform with SSDs. The first stage of the workflow—SamToFastqAndBwaMemAndMba—sustains at close to 100 percent utilization as each workflow task requests 2 threads to run. This translates to running 28 jobs in parallel on the Intel Xeon Platinum 8180 processor platform, and 22 jobs in parallel in the Intel Xeon processor E5-2699 v4 platform. For all the 50 WGS workflows submitted, this first stage completes close to 5.5 days into execution on the Intel Xeon Scalable Platinum 8180 processor platform, while it takes close to 8 days on the Intel Xeon processor E5-2699 v4 platform due to lesser core count.

The next stage—MarkDuplicates—starts executing in parallel (based on resources available) as the first stage starts completing for some workflows. The first drop in CPU utilization happens as the execution enters the phase of the workflow where Samtools Sort and the first few GATK steps start running in parallel (based on resources available for the respective tasks) when the previous stages finish. As observed with latency runs, the Samtools Sort step performs well on SSDs compared to HDDs (based on the memory-per-thread argument for Samtools). For the blue line, the execution that happens between 6-8 days is a combination of Samtools Sort, GATK BaseRecalibrator, GATK ApplyBQSR, and some initial GATK HaplotypeCaller steps. As all the workflows complete their tasks up to HaplotypeCaller, we start seeing sustained utilization of only HaplotypeCaller and MergeVCFs tasks, which only take up 45-50 percent of the utilization, as we are constrained by the memory requirements for this step of the workflow in all three configurations. Comparing the valleys where the combination of workflow tasks execute across all three configurations, the execution of the workflows on HDDs takes longer to reach the sustained utilization for HaplotypeCaller (close to 9 days into execution). This happens because all the stages of the workflow after SamToFastqAndBwaMemAndMba significantly read and write pages to and from the HDDs, which proves to be a costly factor with HDDs. In contrast, the valleys for both the dark blue and orange lines, which compare the Intel Xeon Platinum 8180 processor and the prior-generation Intel Xeon processor E5-2699 v4, are similar and the execution times are only slightly different because these stages are not bound by CPU or memory or SSDs across processor generations. Similar to the latency experiments, the two states of the workflow that significantly speed up with the latest-generation processor—where the higher core count makes a difference—are SamToFastqAndBwaMemAndMba and HaplotypeCaller. In all three configurations, the first of the 50 workflows starts completing a few hours into the sustained 50 percent CPU utilization at the end of their respective colored lines.

Memory Utilization

Evaluating Figure 11, we observe a comparison between the Intel Xeon Platinum 8180 processor with SSDs (blue line), Intel Xeon Platinum 8180 processor with HDDs (dark blue line), and prior-generation Intel Xeon processor E5-2699 v4 with SSDs (orange line). The vertical axis shows the node's maximum memory utilization at close to 256 GB for SamToFastqAnd BwaMemAndMba for the Intel Xeon Platinum 8180 processor and 200 GB for the same step running on the prior-generation Intel Xeon processor E5-2699 v4, out of a total 512 GB available on each of the tested configurations.

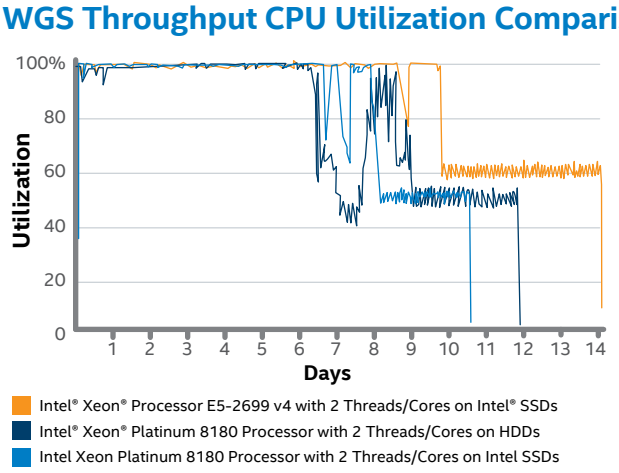


Figure 10. Whole Genome Sequence (WGS) throughput CPU utilization comparison.

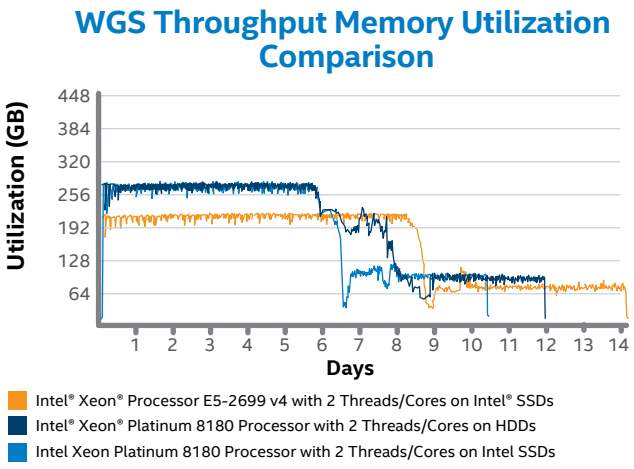


Figure 11. Whole Genome Sequence (WGS) throughput memory utilization comparison.

Per the learnings from the latency experiment, no swap space was made available to the workflow tasks in the HDD experiments to avoid frequent swapping. Because of this, the Intel Xeon Platinum 8180 processor with HDDs configuration's (dark blue line) memory utilization for the Samtools Sort and the GATK stages of the workflow is higher and trending differently compared to the SSD experiments (across both generations). The only stage of the workflow that is bound by memory is HaplotypeCaller, where each job of HaplotypeCaller requests 21 GB of memory, so at a given time only 23 or a maximum of 24 HaplotypeCaller jobs can run. This translates back to the average of 45-50 percent average CPU utilization for both generations of processors. The phase shift in execution time observed for the prior-generation processors compared to the latest-generation processors is consistent as observed with the latency runs as well.

Read/Write Performance

The left side of Figure 12 demonstrates the average read and write operations when executing a load of 50 WGS workflows on a system using the Intel Xeon Platinum 8180 processor with SSDs versus the Intel Xeon processor E5-2699 v4 with SSDs. Analogously, the right side demonstrates the average read and write operations when executing the same load on the Intel Xeon Platinum 8180 processor using HDDs. In both graphs, the vertical axis corresponds to the read/write operations per second as observed on SSDs or HDDs; the horizontal axis corresponds to the number of days elapsed. Note that the right side of Figure 12 has fewer days on the horizontal axis compared to the left side, due to the longevity of the experiment with the Intel Xeon processor E5-2699 v4 with SSDs on the left side. Similar to our latency experiments, we observe that the total number of read/write operations is higher in the first stage of the workflow with SSDs due to the sustained connection capacity of SSDs, but are not visible in these graphs due to the much higher disk activity observed in the later stages.

The stages of the workflow where the major activity happens are between days 6-8 for the Intel Xeon Platinum 8180 processor with SSDs, days 6-9 days for the same processor with HDDs, and days 8-10 for the Intel Xeon processor E5-2699 v4 with SSDs. These periods correspond to the many temporary files read and written to disk in the SortSampleBam and GATK stages. This activity is more noticeable in the left side of Figure 12 because pages are swapped in and out of memory more efficiently, compared to HDDs. We observed frequent reads/writes between days 10-14 in the case of the Intel Xeon processor E5-2699 v4 with SSDs during the HaplotypeCaller stage of the workflow, due to an external process running on the node. However, since the CPU and memory utilization of the stage did not change drastically, and the overall execution time and throughput are within expectations, we present this data. Overall, the higher bandwidth of Intel SSDs helps improve the throughput performance of all the stages of the workflow compared to HDDs, by being able to reach close to 12 million IOPS, which is 2.4x higher than what can be achieved with HDDs.

Conclusion

Advancements in genomic sequencing technology and its continually decreasing cost are driving an unprecedented scale of data that must be processed and analyzed. With BIGstack, Intel and Broad take a major leap in overcoming the processing and storage bottlenecks of the past. As evidenced by our test results, BIGstack is ideal for use in situations involving the analysis of large sets of genomic data, such as running the GATK Best Practices workflow from the Broad Institute. Configured with the latest Intel Xeon Scalable processor and Intel 3D NAND SSDs, BIGstack allows researchers to process up to five WGS per day and more than 100 WES per day, with only a single node. Furthermore, this configuration can boost the individual sample analysis by up to 2.2x at a lower memory cost, compared to a platform based on the prior-generation Intel Xeon processor E5-2699 v4.

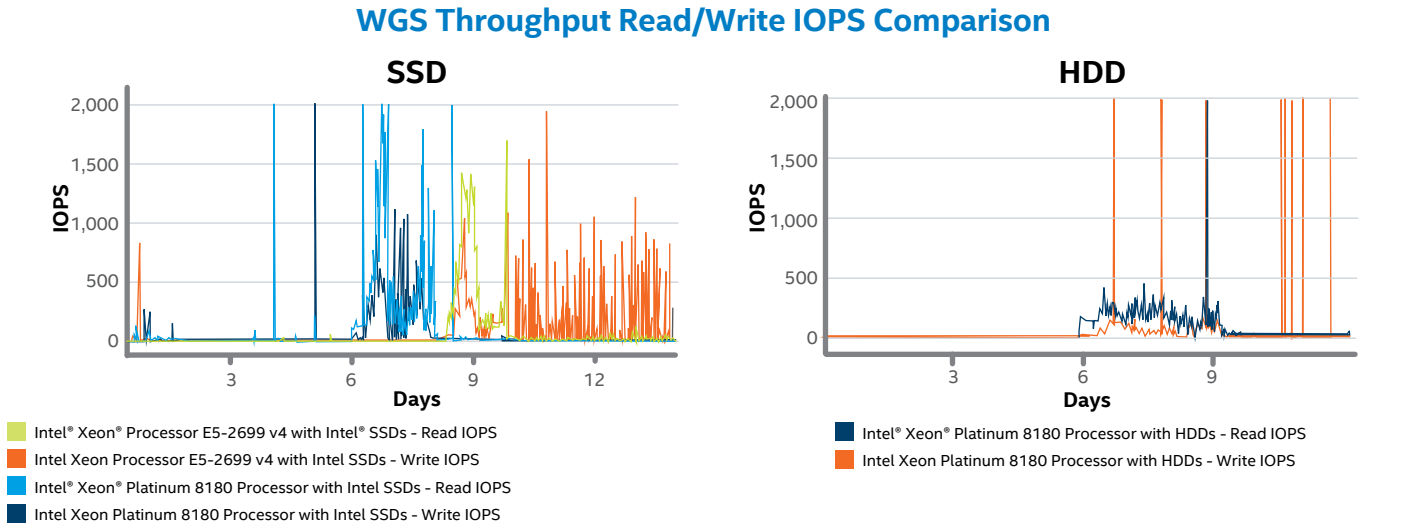


Figure 12. Whole Genome Sequence (WGS) throughput read/write SSD and HDD IOPS comparison.

These results open up multiple exciting avenues for further investigation. Possible future work includes:

- **WES/WGS throughput experiments with Intel Arria 10 FPGAs.** WES and WGS latency experiments were a natural first choice to compare Intel Arria 10 FPGA and Intel AVX performance. In the coming months, we hope to investigate how using Intel Arria 10 FPGAs improves workflow throughput.
- **Smith-Waterman Kernel for FPGAs.** There is an ongoing effort to accelerate the Smith-Waterman algorithm for the Intel Arria 10 FPGA. The GATK HaplotypeCaller tool uses the Smith-Waterman algorithm, so we expect that implementing this kernel in GKL would lead to additional improvements in runtime for both the GATK 3.8 and Hybrid workflows.
- **Intel® Optane™ technology.** Intel SSDs improve latency and throughput performance as a result of their high read/write bandwidth compared to traditional HDDs. The larger bandwidth offered by Intel Optane technology should further extend these gains.
- **Fine tuning memory and core requirements.** The processor cores and memory allocated to each of the workflow stages were tuned primarily so the workflows could run on SSDs or HDDs. There is likely room to further tune memory-bound stages of the workflow, such as HaplotypeCaller.
- **Optimize the pipeline with software tools and piping stages.** Some of the improvements in our results can be attributed to changes in how the workflow stages are arranged and which stages are grouped together. There may be further optimizations that result from regrouping additional workflow stages, as well as additional performance tuning with the release of GATK 4.0.

The results published in this paper show that with optimized hardware and the latest version of BIGstack, analysis of large datasets is now possible at a significantly improved rate.

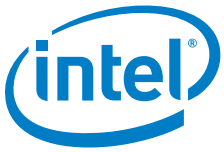
Contact your Intel representative for more information about how Intel can help you and your organization with your genome analysis needs.

Acknowledgements

We would like to thank Geraldine Van der Auwera, Eric Banks, Jose Soto, Megan Shand, Lee Lichtenstein, and Beri Shifaw at the Broad Institute for providing the workflows and datasets used in this work and the ongoing collaboration.

We would like to thank our internal team for their continued support: Paolo Narvaez, Mark Bagley, George Powley, Priya Vaidya, Brian Cremeans, Jason Stark, Jason Yoder, Kushal Datta, Carlos Carrizo, and Ealasaid Haas. We would also like to thank our collaborators across Intel: Mark Laurenz, Vivek Sarathy, Boon Seng Tan, Pedro Vazquez, Mir Ahsan, Mohamed Issa, Richard Yang, Lisa Girouard, and Yujiang Sun.

[1] "The genomics intelligence revolution," techcrunch.com/2017/01/21/the-genomics-intelligence-revolution
[2] Genomic Research by Intel and Broad Institute, intel.com/content/www/us/en/healthcare-it/solutions/genomics-broad-data.html
[3] GATK Best Practices, software.broadinstitute.org/gatk/best-practices/index.php
[4] "Accelerating the Compression and Decompression of Genomics Data using GKL Provided by Intel," intel.com/content/dam/www/public/us/en/documents/white-papers/accelerating-genomics-data-gkl-white-paper.pdf
[5] "GenomicsDB: Storing Genome Data as Sparse Columnar Arrays," intel.com/content/dam/www/public/us/en/documents/white-papers/genomics-storing-genome-data-paper.pdf
[6] Genome Analysis Toolkit, software.broadinstitute.org/gatk
[7] Picard command line tools, broadinstitute.github.io/picard
[8] Burrows-Wheeler Aligner, bio-bwa.sourceforge.net
[9] Samtools, htslib.org
[10] Cromwell, github.com/broadinstitute/cromwell
[11] WDL, software.broadinstitute.org/wdl
[12] Best Practices for Germline SNP and Indel Discovery in Whole Genome and Exome Sequence, software.broadinstitute.org/gatk/best-practices/bp_3step.php?case=GermShortWGS
[13] Computing with HTCondor, research.cs.wisc.edu/htcondor
[14] "Infrastructure for Deploying GATK Best Practices Pipeline," intel.com/content/dam/www/public/us/en/documents/white-papers/deploying-gatk-best-practices-paper.pdf
[15] "Broad Institute teams up with Intel to integrate genomic data from diverse sources and enhance genomic data analytic capabilities," broadinstitute.org/news/broad-institute-teams-intel-integrate-genomic-data-diverse-sources-and-enhance-genomic-data



Accelerate Genomics Research with the Broad-Intel Genomics Stack*

Supplementary Information

Appendix Table of Contents

Appendix A: Datasets

Table A1. Dataset Selection for Whole Genome Sequencing (WGS) at 30x Coverage

Sample	File Used	Size (GB)
NA12878	HK3T5CCXX.8.Pond-492100.unmapped.bam	2.7
	HK3T5CCXX.7.Pond-492100.unmapped.bam	2.7
	HK3T5CCXX.5.Pond-492100.unmapped.bam	2.8
	HK3T5CCXX.4.Pond-492100.unmapped.bam	2.8
	HK3T5CCXX.3.Pond-492100.unmapped.bam	2.8
	HK3T5CCXX.2.Pond-492100.unmapped.bam	2.8
	HK35NCCXX.2.Pond-492100.unmapped.bam	2.8
	HK35MCCXX.8.Pond-492100.unmapped.bam	2.7
	HK35MCCXX.7.Pond-492100.unmapped.bam	2.8
	HK35MCCXX.6.Pond-492100.unmapped.bam	2.8
	HK35MCCXX.5.Pond-492100.unmapped.bam	2.8
	HK35MCCXX.4.Pond-492100.unmapped.bam	2.7

Sample	File Used	Size (GB)
NA12878	HK35MCCXX.3.Pond-492100.unmapped.bam	2.7
	HK35MCCXX.2.Pond-492100.unmapped.bam	2.7
	HK35MCCXX.1.Pond-492100.unmapped.bam	2.7
	HJYN2CCXX.1.Pond-492100.unmapped.bam	2.2
	HJYFJCCXX.8.Pond-492100.unmapped.bam	2.8
	HJYFJCCXX.7.Pond-492100.unmapped.bam	2.8
	HJYFJCCXX.6.Pond-492100.unmapped.bam	2.8
	HJYFJCCXX.5.Pond-492100.unmapped.bam	2.9
	HJYFJCCXX.4.Pond-492100.unmapped.bam	2.8
	HK3T5CCXX.6.Pond-492100.unmapped.bam	2.8
	HK3T5CCXX.1.Pond-492100.unmapped.bam	2.7
	HK35NCCXX.1.Pond-492100.unmapped.bam	2.8

Table A2. Dataset Selection for Whole Exome Sequencing (WES) at 50x Coverage

Sample	File Used	Size (GB)
HG02461	HG02461.unmapped.bam	5.9
HG02462	HG02462.unmapped.bam	7.4
HG02464	HG02464.unmapped.bam	6.0
HG02561	HG02561.unmapped.bam	6.3
HG02562	HG02562.unmapped.bam	5.7
HG02570	HG02570.unmapped.bam	7.2
HG02571	HG02571.unmapped.bam	6.3
HG02582	HG02582.unmapped.bam	5.5
HG02583	HG02583.unmapped.bam	6.2
HG02585	HG02585.unmapped.bam	6.0
HG02588	HG02588.unmapped.bam	6.2
HG02595	HG02595.unmapped.bam	7.0
HG02610	HG02610.unmapped.bam	6.5
HG02613	HG02613.unmapped.bam	5.8
HG02620	HG02620.unmapped.bam	5.9
HG02621	HG02621.unmapped.bam	5.8
HG02623	HG02623.unmapped.bam	6.3
HG02624	HG02624.unmapped.bam	7.0
HG02629	HG02629.unmapped.bam	6.5
HG02634	HG02634.unmapped.bam	6.4
HG02635	HG02635.unmapped.bam	5.8
HG02642	HG02642.unmapped.bam	6.2
HG02645	HG02645.unmapped.bam	6.0
HG02646	HG02646.unmapped.bam	5.9
HG02676	HG02676.unmapped.bam	6.0

Sample	File Used	Size (GB)
HG02678	HG02678.unmapped.bam	5.6
HG02679	HG02679.unmapped.bam	6.0
HG02715	HG02715.unmapped.bam	7.2
HG02716	HG02716.unmapped.bam	5.7
HG02721	HG02721.unmapped.bam	5.7
HG02756	HG02756.unmapped.bam	6.4
HG02757	HG02757.unmapped.bam	6.4
HG02768	HG02768.unmapped.bam	6.0
HG02769	HG02769.unmapped.bam	6.2
HG02771	HG02771.unmapped.bam	6.3
HG02772	HG02772.unmapped.bam	5.8
HG02798	HG02798.unmapped.bam	7.1
HG02799	HG02799.unmapped.bam	5.9
HG02804	HG02804.unmapped.bam	7.1
HG02807	HG02807.unmapped.bam	6.6
HG02808	HG02808.unmapped.bam	6.7
HG02810	HG02810.unmapped.bam	7.2
HG02811	HG02811.unmapped.bam	6.1
HG03027	HG03027.unmapped.bam	6.9
HG03028	HG03028.unmapped.bam	7.5
HG03039	HG03039.unmapped.bam	5.8
HG03040	HG03040.unmapped.bam	7.4
HG03046	HG03046.unmapped.bam	6.7
HG03049	HG03049.unmapped.bam	6.1
HG03258	HG03258.unmapped.bam	6.9

Table A3. Resource Files for Single-Sample Germline Variant Calling Workflow

File Type	File Used	Size
Reference Genome	Homo_sapiens_assembly38.fasta	3.1 GB
dbSNP VCF File	Homo_sapiens_assembly38.dbsnp138.vcf	11 GB
Indels VCF File	Mills_and_1000G_gold_standard.indels.hg38.vcf.gz	20 MB
HapMap VCF File	hapmap_3.3.hg38.vcf.gz	60 MB
Omni VCF File	1000G_omni2.5.hg38.vcf.gz	51 MB
1000 Genome	1000G_phase1.snps.high_confidence.hg38.vcf.gz	1.8 GB
WGS Intervals	hg38_wgs_scattered_calling_intervals.txt	7.5 KB
Exome Intervals	hg38_es_scattered_calling_intervals.txt	6.9 KB

Appendix B: WES GATK* v3.8 Single-Sample Germline Variant Calling Workflow

The Single-Sample Germline Variant Calling workflow for WES implements data pre-processing and initial variant calling (gVCF generation) according to the GATK* Best Practices (June 2016) used for germline SNP and Indel discovery in human sequencing data. This workflow was implemented with GATK v3.8 tools, and tested with WES only. The workflow starts with unmapped BAM files and runs the different tools from SamToFASTQ and BWA-Mem to GATK HaplotypeCaller and Picard* MergeVCFs for all samples. Figure B1 describes the sequence of steps used for this workflow.

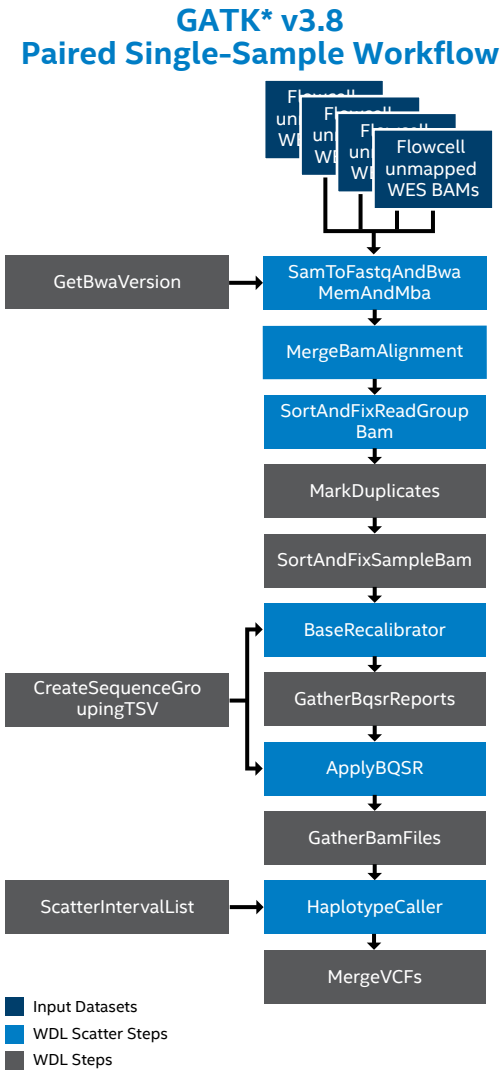


Figure B1. GATK* v3.8 Single-Sample Germline Variant Calling workflow

Appendix C: WGS GATK* Hybrid Single-Sample Germline Variant Calling Workflow

The WGS Single-Sample Germline Variant Calling workflow is a Hybrid workflow for WGS, meaning that it uses multiple GATK* versions (GATK v3.8 for Haplotype Caller and GATK v4.0 beta.5 for all other GATK tools), and it makes several optimizations to the GATK v3.8 Single-Sample workflow. These improvements include piping the output of SamToFastqAndBwaMem to MergeBamAlignment, thus saving a costly write and read from disk. In addition, the two sorting steps in the GATK v3.8 Germline Variant Calling workflow are consolidated into a single Samtools* Sort step. Figure C1 describes the sequence of steps used for this workflow.

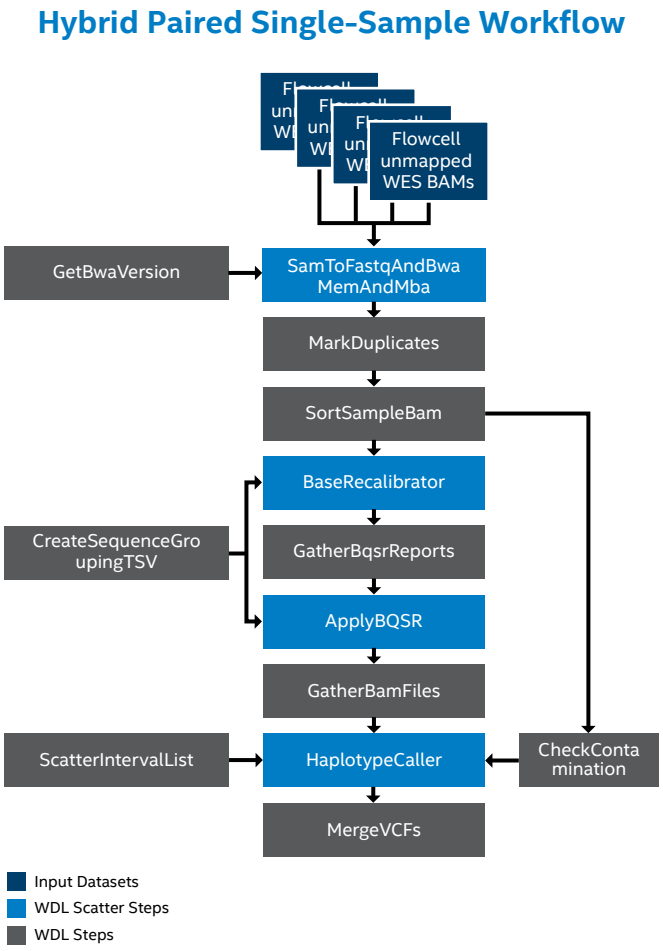


Figure C1. Hybrid Single-Sample Germline Variant Calling workflow. All GATK* steps except HaplotypeCaller use GATK v4.0 beta.5. HaplotypeCaller uses GATK v3.8

Appendix D: Configuration Adjustments

In addition to the hardware configuration listed in the Hardware section, we recommend verifying that the system has the latest available BIOS and making the following changes prior to benchmarking system performance:

- Verify the BIOS version is higher than 1.0004:
  - `dmidecode | grep -A 2 BIOS | grep -i version`
  - Expected output: Version: SE5C620.86B.00.01.0004.071220170215
- Enabling turbo mode in the system BIOS
- Set the Linux* frequency governor to performance mode:
  - `cpupower -c all frequency-set -g performance`
- If the target system is configured with hard drives, follow the steps below to avoid swap thrashing:
  - Paste the following text into `/etc/cgconfig.d/condor`

```
group condor
{
        cpuset {}
        cpu {}
        cpuacct {}
        memory {
          memory.swappiness = 0;
        }
        devices {}
        freezer {}
        net_cls {}
        blkio {}
        perf_event {}
}
```

  - `systemctl restart cgconfig`
  - `echo 0 > /sys/fs/cgroup/memory/condor/memory.swappiness`
- The Java* version used for all experiments is `"OpenJDK build 1.8.0 151-b12"`

If you encounter any issues with the above commands, please contact your assigned Intel representative.

Appendix E: Details on Tools and Technologies

CROMWELL*

Users submit workflows to the Cromwell* execution engine. These workflows are composed in the form of Workflow Description Language (WDL) files, and make use of numerous tools optimized by Intel, including the Genomics Kernel Library* (GKL*) and GenomicsDB*.

HTCONDOR*

When a user submits a workflow, Cromwell directs HTCondor* to schedule the workflow's tasks using available resources. HTCondor handles the allocation of tasks, making sure that tasks are scheduled based on application requirements. Cromwell and HTCondor also make it possible to run multiple workflows concurrently, such as a batch of processes.

GENOMICSDB*

GenomicsDB is a data store for variants based on the TileDB* array storage manager. TileDB is a system for efficiently storing, querying, and accessing sparse and dense matrix/array data. It is developed by researchers at the Intel® Science and Technology Center for Big Data.

Variant data, such as the gVCF files produced by the Single-Sample Germline Variant Calling workflow, is sparse by nature (sparse relative to the whole genome), which makes TileDB an excellent fit. GenomicsDB adds on to the TileDB platform, making it even more specialized and maximizing efficiency for handling genomic data.

GenomicsDB stores variant data in a two-dimensional TileDB array where:

- Each column corresponds to a genomic position (chromosome and position).
- Each row corresponds to a sample in a VCF ("CallSet" in GA4GH terminology).
- Each cell contains data for a given sample or CallSet at a given position; data is stored in the form of TileDB cell attributes.
- Cells are stored in column major order; this speeds up accessing cells with the same column index (that is, data for a given genomic position over all samples).
- Variant interval/gVCF interval data is stored in a cell at the start of the interval. The END is stored as a cell attribute. When queried for a given genomic position, the query library performs an efficient sweep to determine all intervals that intersect with the queried position.

GenomicsDB can be configured to store variant data across multiple partitions of an array. All the data belonging to one partition of an array is stored in a single file system. Thus, by creating multiple partitions, users can store data across multiple hosts or nodes in a cluster. Array partitioning is useful when the data to be stored and queried is very large and cannot fit within a single machine or node. Alternatively, the user might wish to store array partitions in different nodes so that downstream queries and analysis can be run in a distributed manner for scalability or performance.

For more details on GenomicsDB, please see GenomicsDB: Storing Genome Data as Sparse Columnar Arrays.

GENOMICS KERNEL LIBRARY*

The Genomics Kernel Library (GKL) is a collection of common, compute-intensive kernels used in genomic analysis tools. Intel and the Broad Institute worked together to identify these kernels in GATK. Experts across Intel optimized the kernels for Intel® architecture and released them in the GKL. Intel and Broad worked together to update GATK to use the GKL. The kernels provided by the GKL include those listed in Table E1.

Table E1. Genomics Kernel Library* (GKL*) Kernels

Kernel	Description	Tools That Use It
PairHMM	Advanced Vector Extensions (Intel® AVX/AVX-512) and Intel® Arria® 10 FPGA version of the PairHMM algorithm, which is used to perform pairwise alignment of reads vs. haplotypes in GATK* HaplotypeCaller and MuTect2	HaplotypeCaller and MuTect2
ISA-L igzip compression	High-performance level 1 compression dynamically optimized for genomic data	All GATK tools that write BAM files
OTC zlib compression	Optimized zlib compression for levels 2 through 9	All GATK tools that write BAM files
INFLATE decompression	Optimized decompression for data compressed in the DEFLATE format	All GATK tools that read BAM or block-gzipped VCF files

The GKL currently supports optimized native libraries for Linux* and OSX*, and contains Java* wrappers for GATK and HTSJDK. The compression and decompression kernels in GKL provide a high level of acceleration for compression and decompression of BAM and block-gzipped VCF file formats used within the GATK suite. For end users, these are integrated into GATK using the precompiled release on Maven Central.

GATK also supports standard C/C++ applications through an optimized libz.so, a zlib drop-in replacement for optimized compression in a native C/C++ framework. This is also open source and released on GitHub as part of the GKL.

Table E2 provides information on how to optimize the GKL settings for running the workflows described in the Workflows and Data section.

For more information on the GKL, please see Accelerating the Compression and Decompression of Genomics Data using GKL Provided by Intel.

Table E2. Optimized Genomics Kernel Library* (GKL*) Settings

Kernel Setting	Default	Optimized	Summary
--pair_hmm_implementation	FASTEST_AVAILABLE	System dependent Intel benchmarking used VECTOR_LOGLESS_CACHING or VECTOR_LOGLESS_CACHING_FPGA_EXPERIMENTAL	By default, GATK* will select the fastest available PairHMM implementation on the system.
--nativePairHmmThreads	4	1	The number of threads should be tuned based on system configuration and expected workload.
--useDoublePrecision	false	false	Set to true to match the results of Java* PairHMM with about half the performance of single precision.
--bam_compression	1	1	Sets compression level (1 is low, 9 is high). Compression level 1 provides much higher performance with slightly larger output.