WHITE PAPER

Accelerating the Compression and Decompression of Genomics Data using GKL Provided by Intel

Authors
James Guilford, George Powley, Greg Tucker, Priya Vaidya, Intel Corporation
Louis Bergelson, Lee Lichtenstein, David Roazen, Broad Institute

Genomics Kernel Library (GKL) is provided by Intel as open source to optimize the performance of genomics applications on Intel® architecture-based system hardware. GKL offers up to a 2.7x performance improvement for compression and up to 2x for decompression.[1]

Intel provides GKL to expose performance optimizations for Intel® architecture to developers of genomics applications, by means of optimized native libraries for Linux* and Mac OS X*. GKL includes Java* wrappers for the GATK (Genome Analysis Toolkit) and HTSJDK (High Throughput Sequencing JDK) developed by the Broad Institute of MIT and Harvard. As illustrated in Figure 1, GKL occupies the space in the overall stack between Intel architecture-based hardware and genomics applications. As a library of optimized, open-source software components, GKL provides the means for those applications to fully take advantage of the hardware capabilities. Once GKL is installed, its usage and performance benefits in GATK3 and GATK4 are automatic and seamless.

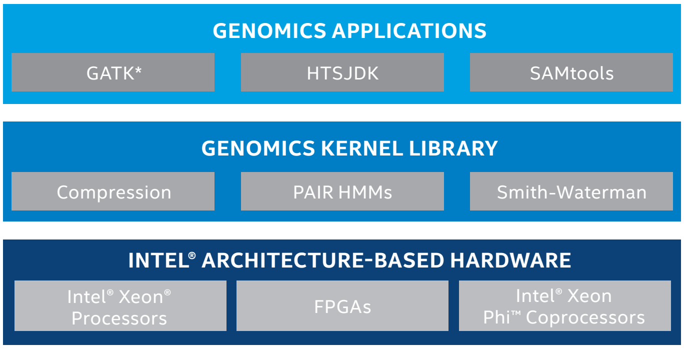

Figure 1. GKL in the genomics solution stack.

This paper discusses the compression and decompression component of GKL, including technical details about the acceleration algorithms and technologies used, their implementation details, and incorporation of these capabilities into GATK for delivery within end-user applications. It also provides benchmarking results to compare GKL implementations with other Java implementations, highlighting the benefits of using GKL.

CPU-Accelerated Compression and Decompression with GKL

As shown in Figure 1, accelerating compression and decompression routines for BAM files and other formats used within GATK is a significant aspect of the opportunity for performance improvement using GKL in genomics applications. GKL provides up to an approximately 2.7x performance improvement for compression over the standard Java implementation with minimal loss of compression ratio and up to about 2x performance improvement for decompression.

In addition to native support for Java applications, GATK also provides native C/C++ support. It includes **libz.so**, a drop-in replacement for **zlib** that provides optimized compression and decompression in a native C/C++ framework. This component is also open source and released on Maven Central as part of GKL.

For compression level 1 functions, GKL provides a fast, DEFLATE-compatible compression routine. DEFLATE is a widely used binary compression algorithm that is the basis of **zlib**, **gzip**, and **zip**. This Intel® Intelligent Storage Acceleration Library compression implementation is designed to provide higher performance than **zlib-1** with only a small sacrifice in compression ratio. It is well suited to high-throughput usages such as genomics and enterprise storage applications. The current implementation uses dynamic Huffman, which reads a portion of the decompressed file to generate optimized tables. There is no need to provide static tables.

For compression beyond level 1, GKL uses a set of 13 patches to **zlib** created by the Intel Open Source Technology Center, which sharply improve DEFLATE performance. These include tuning and other changes such as medium and quick DEFLATE strategies, a faster hash function with Intel® Streaming SIMD Extensions (Intel® SSE) 4.2 support, PCLMULQDQ-optimized CRC folding, and Intel SSE2 hash shifting.

Benchmarking Results

To demonstrate the potential benefit of GKL to compression and decompression performance, an Intel test team compared GKL-optimized versions of Java Deflater* and Java Inflater* with those commonly used by GATK and other genomics applications. Testing was performed on a server based on the Intel® Xeon® processor E5-2699 v4, running sample BAM files (~300 MB) as input and using GKL release 4.3. Measurements were captured using the timing functions provided within Linux.

Across all compression levels, optimization of both Java Deflater and Java Inflater using GKL provides significant performance improvement, as shown in Figure 2.[1] For compression, GKL optimization of Java Deflater provides in excess of 2.7x performance improvement at compression level 1, and in the range of approximately 1.3x to 1.5x improvement at compression levels 2 through 9. For decompression, results from GKL optimization of Java Inflater range from nearly 1.7x to more than 2x performance improvement, with the greatest improvements at compression levels 3 and above.

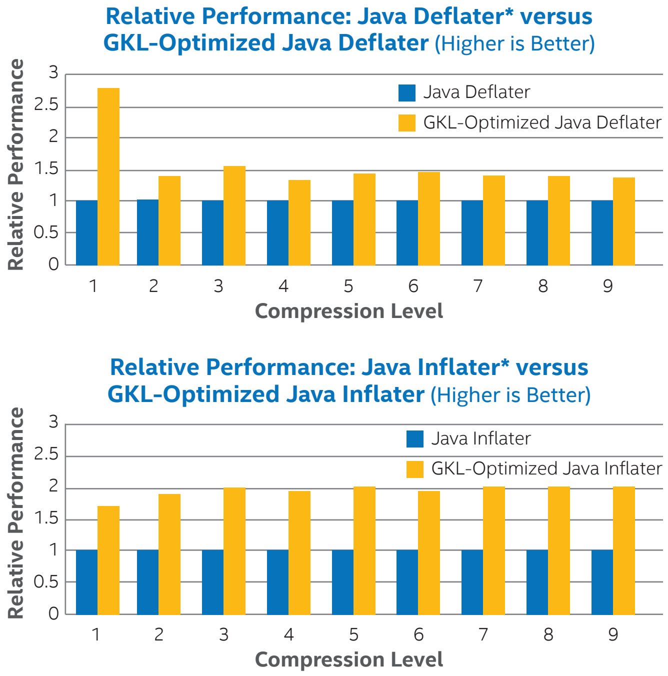

Figure 2. Performance comparison with and without GKL optimization.[1]

The relationship between these performance improvements and the data-compression ratio for levels 1, 5, and 9 is illustrated in Figure 3.[1] At all three compression levels, GKL produces faster completion time for the compression task, with minimal relative decrease in compression ratio. For example, at compression level 1, the GKL-optimized Java Deflater offers a performance increase of approximately 2.8x over the non-optimized version, with an increase in compressed file size of only about 1.8 percent.

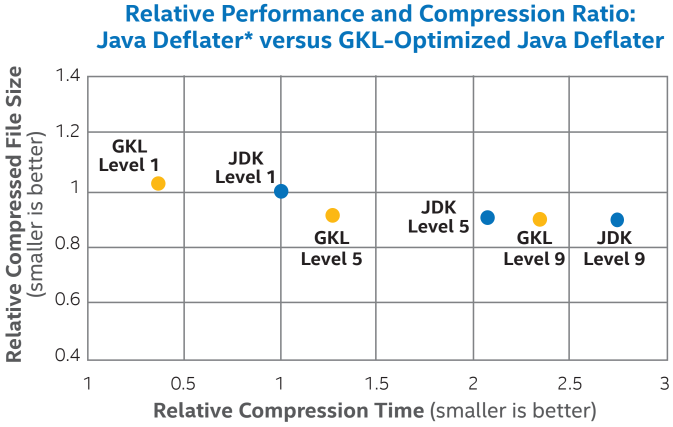

Figure 3. Relationship between GKL effects on performance and compression ratio.[1]

Conclusion

GKL offers significant performance improvements to genomics applications. Optimization of compression and decompression routines illustrates the potential for accelerating existing as well as future research software. Ongoing work will measure results across other parts of GKL and on other types of Intel architecture-based systems, with the ultimate goal of enabling faster completion of more complex genomics tasks on available hardware.

Download GKL at
https://github.com/Intel-HLS/GKL

or contact **george.s.powley@intel.com**
or **priya.n.vaidya@intel.com**

For more information about Intel in genomics, visit
www.intel.com/broadinstitute

[1] System under test: 2x Intel® Xeon® processor E5-2699 v4, 256 GB RAM, Intel® SSD DC P3700 Series (2 TB), sample input BAMfile (~300 MB, provided by the Broad Institute), GKL release 4.3, Java Deflater* module, Java Inflater* module.

Software and workloads used in performance tests may have been optimized for performance only on Intel® microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.