Performance optimization of Broad Institute GATK Best Practices on IBM reference architecture for healthcare and life sciences

IBM's commitment to enhance performance

Overview

The Genome Analysis Toolkit (GATK) Best Practices from the Broad Institute [1] has been widely adopted by the genomics community to perform variant discovery analysis of next-generation sequencing (NGS) data. A 30 times coverage of the whole human genome can take days to process using GATK Best Practices pipeline [2]. This paper describes how IBM has significantly accelerated the workflow on IBM reference architecture for healthcare and life sciences. It demonstrates that the GATK workflows can take advantage of the simultaneous multithreading (SMT) feature of IBM® POWER8® by parallelization of the GATK workflow. With the optimization on IBM's reference architecture, it takes approximately 10 hours to complete GATK Best Practice pipeline for germline variant detection with 30 times coverage of the whole human genome using the GRCH37 reference genome and approximately 13 hours using the GRCH38 reference genome. These timings represent a significant speedup compared to the published Intel® results [2].

Overview

Challenge

Customers need faster turnaround time for processing the GATK best practices pipeline from the Broad Institute.

Solution

IBM has optimized performance of GATK Best Practices pipeline on the IBM POWER8 platform by taking advantage of unique features of IBM

IBM reference architecture for healthcare and life sciences

The IBM reference architecture for healthcare and life sciences [3] is designed to address the common requirements from organizations pursuing genomics, personalized medicine, and other big data initiatives in biomedical research. In the era of diverse workloads, diverse infrastructure needs, and diverse versions of application frameworks, it is important to build an infrastructure that can perform to meet the institution's needs, meanwhile avoiding purpose built compute clusters. The ability to use fewer compute resources using intelligent policy-based workload, data management tools, and computing acceleration are the key aspects of supporting a dynamic research environment. The foundational elements of the reference architecture include IBM Spectrum™ Computing for workload management, IBM Spectrum Scale™ and IBM Elastic Storage Server (ESS) for data storage and management, and IBM POWER® for computational acceleration.

- IBM Spectrum Computing dynamically allocates computational tasks across compute servers in a manner that is transparent to the users.
- IBM Spectrum Scale, formerly known as the IBM General Parallel File System (IBM GPFS™) has been adopted by leading genomics centers, medical institutions, and pharmaceutical companies to address I/O performance, data management, and data integration challenge. IBM ESS is a storage implementation of IBM Spectrum Scale software.
- The IBM POWER8 processor-based servers, which are integrated with IBM Spectrum Computing and IBM Spectrum Scale, can enable computational acceleration. This is achieved through multithreading, large memory bandwidth, and large cache size offered by the IBM POWER8 processors which address all aspects of biomedical research workloads.

Features of IBM POWER8, the IBM OpenPOWER offering, are designed to support compute and memory-intensive workloads. The POWER8 processors can employ up to eight hardware threads per core and advanced memory subsystems are available to achieve leading edge performance. The POWER8 processor makes use of a large number of on- and off-chip memory caches to reduce memory latency and generate very high bandwidth for memory and system I/O. Additional features that have been designed to augment the performance of POWER8 systems include coherent Accelerator Processor Interface (CAPI) (to allow low latency access of accelerators) and NVLink (to provide low latency communication between GPU/GPU and GPU/CPU).

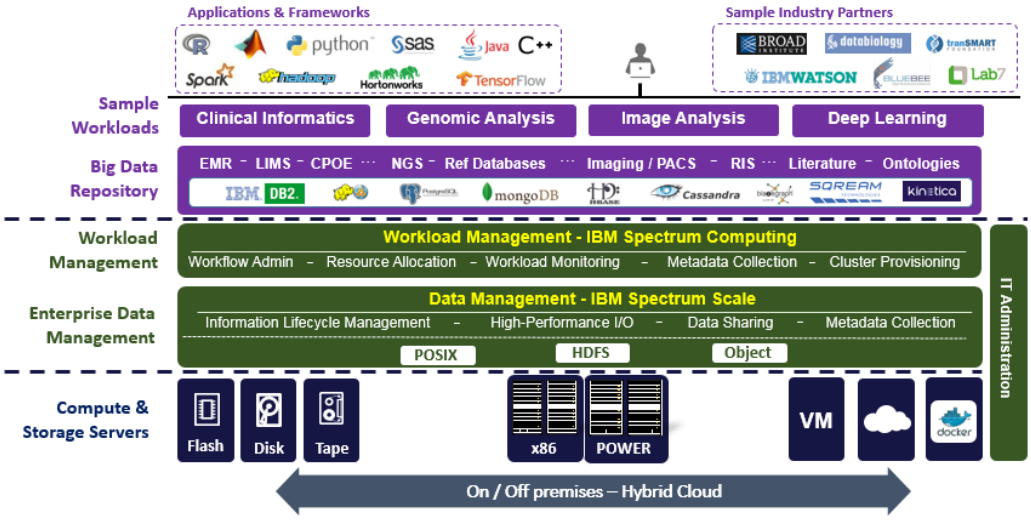

Figure 1: IBM reference architecture for healthcare and life sciences

GATK Best Practices

The GATK Best Practices are a set of applications within a workflow for variant discovery analysis of both germline and somatic genomes with GATK recommended by Broad Institute [4]. This document describes the performance optimization of the workflow for germline single nucleotide polymorphisms (SNPs) and Indel discovery, as shown in Figure 2.

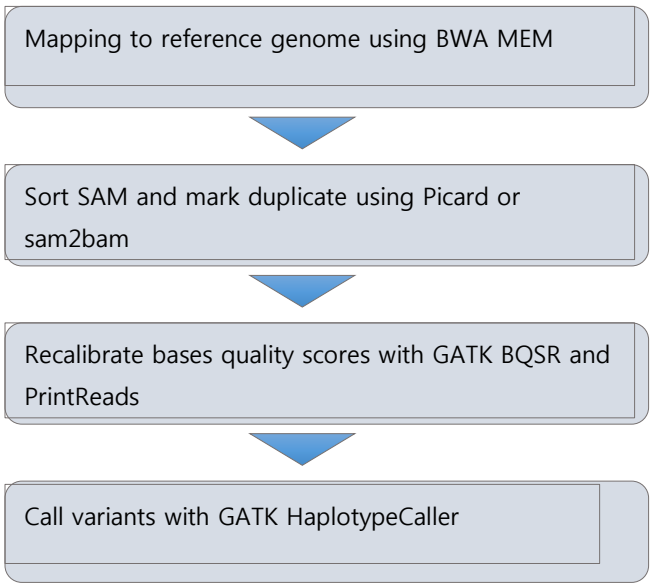

Figure 2: Workflow for germline SNP and Indel discovery

The input data, reference genomes, and reference VCF files are listed in Table 1. The input data Solexa-272221 was provided by Broad Institute. The reference genome and reference variant files were downloaded from resource bundle FTP site from Broad Institute (ftp://gsapubftp-anonymous@ftp.broadinstitute.org/bundle/). Note that users need to create an account before they can download the files. Both GRCH37 and GRCH38 reference genomes were used for benchmarking. The applications and libraries used in this benchmark are shown in Table 2.

Reference genome	Reference VCF	Input data
GRCH38 (hg38) Homo_sapiens_assembly38.fasta	1000G_phase1.snps.high_confidence.hg38.vcf	Solexa-272221
GCH37 (hg19) human_g1k_v37_decoy.fasta	1000G_phase1.indels.b37.vcf dbsnp_138.b37.vcf	Solexa-272221

Table 1: Reference genomes, reference variant files, and input data

Tool	Version	Link
BWA	0.7.15-r1142	https://biobuilds.org/tools-in-biobuilds/biobuilds-2016-11/
Sam2bam	1.1.0	https://github.com/OpenPOWER-HCLS/sam-to-bam
GATK	3.6.0	https://software.broadinstitute.org/gatk/download/
pairHMM library		https://github.com/OpenPOWER-HCLS/GATK3-VectorPairHMM

Table 2: Applications and application version

Configuration of benchmark system

An IBM Power® System S822LC (8335-GTA) server (based on the POWER8 processor technology) connected to ESS GL4 with Spectrum Scale through Mellanox FDR InfiniBand® Switch was used for the optimization work. The detailed configuration of the benchmark system is shown in Table 3.

System configuration	
Processor	3 GHz POWER8
Number of sockets	2
Cores per socket	10
SMT	8 threads per core
Memory	1 TB RAM DDR3
Local drive	Two 3.84 TB SSD Samsung MZ7LM3T8HCJM
Network storage	ESS GL4 with Spectrum Scale 4.2.1-1
Network	Mellanox MT27700 ConnectX-4 IBcard with a Mellanox FDR InfiniBand Switch
Operating system	RHEL 7.2

Table 3: Benchmark system

Installation of tools

You can download all the tools from the links listed in Table 2. This section provides detailed instructions about installation of each tool and running scripts.

Install BWA

You can download POWER8 optimized BWA from BioBuilds, which is a curated and versioned collection of open source bioinformatics tools for genomics [5]. The test team installed BioBuilds using the following instructions.

```
wget
https://repo.continuum.io/miniconda/Miniconda2-
4.2.12-Linux-ppc64le.sh

bash Miniconda2-4.2.12-Linux-ppc64le.sh -p
$PWD/miniconda

./miniconda/bin/conda update conda

./miniconda/bin/conda create -c biobuilds -p
$PWD/biobuilds biobuilds=2016.11
```

Install sam2bam

Sam2bam is a high-throughput software tool framework developed by IBM and is highly optimized based on the IBM Power Architecture® [6]. It can significantly accelerate the conversion of file format from SAM to BAM, sorting and marking the duplicate with the same final results as running Picard SortSam and Picard MarkDuplicates. Installation requires a patch command and standard development tools such as `make` and GNU GCC. The binary of sam2bam resides in the *build* directory.

```
mkdir sam2bam

pushd sam2bam

wget https://github.com/OpenPOWER-HCLS/sam-to-
bam/raw/master/build.sh

bash build.sh

popd
```

Install GATK

The GATK .jar file, *GenomeAnalysisTK.jar*, is downloaded from the Broad Institute site using the link in Table 2. The pairHMM native library optimized for POWER8, libVectorLoglessPairHMM.so (which significantly speeds up variant calling from HaplotypeCaller walker) can be downloaded from OpenPOWER github listed in Table 2. The library can be loaded at GATK runtime by setting the library path using the following command:

-Djava.library.path=/location/of/libVectorLoglessPairHMM.so (for example, the current directory).

Installation of the library needs header files of Java™.

```
git clone https://github.com/OpenPOWER-
HCLS/GATK3-VectorPairHMM
pushd GATK3-VectorPairHMM/
make JRE_HOME=$JAVA_HOME/jre (e.g. make
JRE_HOME=/usr/lib/jvm/java/jre)
cp libVectorLoglessPairHMM.so ..
popd
```

Install optimization scripts

The test team installed the optimized scripts on the current directory [7]. The following example shows the installation steps for users who use GRCH38 (hg38) reference files on a 20-core node. To run scripts on the 16-core node, **a script for 16 cores should be used**. Also, to run the scripts for GRCH37, **B37** is used instead of **hg38**.

```
git clone https://github.com/OpenPOWER-
HCLS/GATK3-optimized-pipeline
rsync -a GATK3-optimized-pipeline/20-
cores/hg38/ .
```

Run GATK Best Practices pipeline using optimized scripts

The test team optimized the GATK Best Practice pipeline by using sam2bam to accelerate SortSam and MarkDuplicates and parallelizing the workflow using scripts that optimize load balancing.

Mapping with BWA-MEM

The test team ran BWA-MEM [8] with 160 threads to take advantage of the SMT8 mode of the POWER8 processors. The example script is run1.BWA.

```
run1.BWA    > run1.BWA.log   2>&1
```

Running SortSam and MarkDuplicates

Sam2bam is used to replace Picard SortSam and MarkDuplicates. Sam2bam is a high-performance framework for NGS data preprocessing [6]. The internal architecture of sam2bam is pipelined and multithreaded to fully utilize all the threads on a single system node. sam2bam can convert the file format from SAM to the compressed BAM. It is possible to add plug-ins to sam2bam for additional analysis. The test team used sam2bam for marking duplicate alignments and removing duplicate alignments while converting input SAM files to the corresponding sorted BAM files. You can run sam2bam at both memory mode and storage mode.

You can achieve optimal performance with no read access to the external storage while compressing and writing BAM records to an output file when running at memory mode. Sam2bam at storage mode requires smaller memory footprint and the performance is limited by the performance of external storage. The example script is run23.SSMD. The performance of sam2bam can be further improved by using a hardware card that provides FPGA-based zlib acceleration and a compression accelerator plug-in [9].

```
run23.SSMD  > run23.SSMD.log 2>&1
```

Running BaseRecalibrator, PrintReads, and HaplotypeCaller

Contig-level parallelization, in addition to the multithreading feature of GATK (-nct option) [10] is used to take advantage of a massive number of threads on the POWER8 system with SMT8 mode. For contig-level parallelization, the test team ran multiple instances of GATK while specifying contigs of human reference genomes (for example, chromosome 1). Each GATK instance processes only alignments for specified contigs. For example, if you run 10 GATK instances for chromosomes 1 through 10, ten processes can be run in parallel.

To further optimize the workflow, the team sorted the execution order of GATK instances based on offline profiles of the runtime for processing contigs and started GATK instances in the sorted order. This approach can minimize the chance of increasing the total runtime of a given step by making a longest GATK instance wait for others. If you run GATK instances for multiple contigs in parallel for a given GATK Best Practices step, the total runtime of the step is the runtime of a GATK instance that shows the longest runtime. The runtime of GATK instances tends to be longer as the sizes of the contigs are larger [11].

More importantly, you can optimize the total runtime of a GATK Best Practices step by reducing the runtime of a GATK instance that shows the longest runtime. The test team split the processor cores into two core groups: *highway* and *ordinary*. GATK instances that are scheduled on the highway core group can run faster than the ones on the ordinary core group. The difference between highway and ordinary core groups is the number of active hardware threads per core to which software threads of GATK instances are bound to. The highway core group uses fewer active hardware threads than the ordinary core group. The number of cores each core group has and the number of active hardware threads each core has depends on the GATK commands (`BaseRecalibrator`, `PrintReads`, and `HaplotypeCaller`) and the number of GATK instances that are executed in parallel. For example, the highway core group has four cores and uses two active hardware threads per core while the ordinary core group has 16 cores and uses four active hardware threads for `BaseRecalibrator`. As the number of running GATK instances is decreased, typically when a GATK step is reaching a final phase, the total number of required hardware threads is also decreased. To adjust to such changes in the number of required hardware threads, the parameters of each core group (the number of cores and the number of active hardware threads per core) are dynamically changed to use the processor cores efficiently.

The longer GATK instances are scheduled on the highway core group, and the other instances are scheduled on the ordinary core group by using taskset, a tool that is available on Linux®. Suppose that the number of active hardware threads of the highway core group is N, GATK instances that show longer remaining times than the others and have N active threads are scheduled on the highway core group. To identify the longer GATK instances, the remaining times of GATK instances are periodically monitored. GATK instances show the remaining times as their informational messages every 30 seconds. The GATK instance that shows the longest remaining times is the GATK instance to be optimized. The example scripts that run

`BaseRecalibrator`, `PrintReads`, and `HaplotypeCaller` and optimize them are run5.BR, run6.PR and run7.HC. These scripts generate multiple VCF files because of contig-level parallellization. The VCF files can be merged into a single VCF file for further analysis. The example script that merges the resulting VCF files is mergeVCF.sh.

```
run5.BR     > run5.BR.log     2>&1
run6.PR     > run6.PR.log     2>&1
run7.HC     > run7.HC.log     2>&1
mergeVCF.sh
```

Results

The GATK Best Practices pipeline was run with IBM's optimized tools, libraries, and parallelization script. The input file Solexa-272221, a 30-time coverage of the whole human genome was run against both GRCH37 and GRCH38 reference genomes. The total runtime is 9.85 hours using the GRCH37 reference genome and 12.51 hours using the GRCH38 reference genome. The runtime for each step of the pipeline is shown in Table 3.

Reference genome	GRCH37 (hg19) (hours)	GRCH38 (hg38) (hours)
BWA-MEM	4.54	6.56
SortSam and MarkDuplicates	0.18	0.22
GATK BaseRecalibrator	1.18	1.15
GATK PrintReads	1.43	1.50
GATK HaplotypeCaller	2.52	3.08
Total runtime	**9.85**	**12.51**

Table 4: Performance of GATK Best Practice pipeline for both GRCH37 and GRCH38 on POWER8 and ESS

The test team also compared the performance of GATK Best Practices pipeline on POWER8 and ESS and published performance number from previous work on an Intel® Xeon® based platform [2]. Both evaluations used the same input files, reference files, and the same options for software tools. The test team however used the latest versions of tools and workflow recommended by Broad Best Practices. The previous work used GATK 3.4 and BWA 0.7.12 while the test team's system used GATK 3.6 and BWA 0.7.15. For SortSam and MarkDuplicates steps, the team used sam2bam tool, which produces the same final results as Picard SortSam and Picard MarkDuplicates. For Base Quality Score Recalibration (BQSR), the test team used BaseRecalibrator walker with both SNP and Indel VCF reference files as recommended by the latest Broad Best Practices. The previous work used RealignerTargetCreator and IndelRealigner with Indel VCF and BaseRecalibrator with SNP VCF file to accomplish the same BQSR step.

As shown in Table 4, it took approximately 36 hours to process Solexa-272221 on 36 cores Intel Xeon-based platform while it took approximately 10 hours to process the same data set using IBM's optimized workflow and latest Broad's best practice on 20 cores POWER and ESS. If we evaluate the amount of system resources

reserved for the pipeline by using a metric of core hours (number of cores in the system x elapsed time), IBM's solution consumes 6.6x less system resources. As shown in Table 4, it took 197 core hours on POWER platform while it took 1296 core hours on Intel Xeon-based platform to process the same data.

Workload	Elapsed time (hours)		Core hours	
	POWER8 with 20 cores	x86 with 36 cores	POWER8 with 20 cores	x86 with 36 cores
BWA MEM	4.54	3.85	90.8	138.6
SortSam and MarkDuplicates	0.18	12.35	3.6	444.6
GATK BaseRecalibrator	1.18	6.09	23.6	219.2
GATK PrintReads	1.43	7.47	28.6	268.9
GATK HaplotypeCaller	2.52	6.32	50.4	227.52
Total	**9.85**	**36**	**197**	**1296**

Table 4. Comparative performance benchmark of GATK Best Practice pipeline on IBM' POWER versus published data on an x86 platform (Intel Xeon processor E5-2699 v3 @ 2.30 GHz, 256 GB RAM) [2]

Conclusion and future work

The advances in genomics sequencing over the past two decades has reduced the cost of whole genome sequencing (WGS) significantly. Biomedical research centers need to process the raw sequencing data to achieve a result that is medically actionable. Broad Best Practices pipeline is one of most commonly used pipelines to process raw sequencing data to variant files. Because it is highly desirable to reduce the runtime of GATK, the test team approached this challenge by taking advantage of SMT of the POWER8 processor and by parallelization of tools and workflows, thus reducing the processing time from days to hours with the same accuracy.

Broad is currently developing GATK4, which simplifies and streamlines the Best Practice pipeline and enables massively parallel execution on local clusters or in the cloud using Apache spark. The test team is working to further optimize the future GATK4 with SPARK on the POWER8 platform and ESS. The preliminary results suggest that SPARK, which performs about two times faster on POWER8 than x86 [12], can significantly improve the performance of GATK4. The test team might also further enhance the parallelization of GATK4 workloads with IBM Spectrum Computing Load Sharing Facility (LSF), and Conductor with SPARK. IBM Spectrum LSF and Conductor with SPARK can manage workloads for high-performance computing including genomic data analysis. In the LSF enabled environment, multiple instances of GATK for processing contigs run in parallel and can be managed by LSF.

The team believes that *precision* medicine, which has been a focus of national attention around the world, has far reaching implications that affect us all. IBM's optimization of the GATK pipeline on IBM POWER8 is an enabler of these lofty goals. People must gain better insights into the biological, environmental, and genetic factors to gain understanding as to why individuals respond differently to prevention of disease and its treatment. *Precision medicine* is an emerging approach for disease treatment and prevention. IBM's ongoing

commitment to optimization can help process the data that leads to the understanding of individual variability in environment, lifestyle, and genetic factors.

Acknowledgement

Special thanks to Dr. Geraldine Van der Auwera from Broad Institute who not only provided the dataset used in this study but also insight and expertise on the GATK Best Practices pipeline that greatly assisted the team's understanding of the workloads and made the performance optimization possible. The team would also like to thank Janis Landry-Lane for helpful discussions and comments to improve the paper.

References

[1] GATK Best Practices https://software.broadinstitute.org/gatk/best-practices/index.php
[2] Infrastructure for GATK Best Practices Pipeline Deployment http://www.intel.com/content/www/us/en/healthcare-it/solutions/documents/deploying-gatk-best-practices-paper.html
[3] IBM reference architecture for healthcare and life science https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=TSW03536USEN&
[4] Broad Institute (2017) Best Practices for Germline SNP & Indel Discovery in Whole Genome and Exome Sequence https://software.broadinstitute.org/gatk/best-practices/bp_3step.php?case=GermShortWGS
[5] BioBuilds. https://biobuilds.org/
[6] Sam2bam: High-Performance Framework for NGS Data Preprocessing Tools https://github.com/OpenPOWER-HCLS/sam-to-bam
[7] Optimized parallel execution of GATK3 for OpenPOWER https://github.com/OpenPOWER-HCLS/GATK3-optimized-pipeline
[8] Fast and accurate short read alignment with Burrows–Wheeler transform https://www.ncbi.nlm.nih.gov/pubmed/19451168
[9] Sam2bam: High-Performance Framework for NGS Data Preprocessing Tools. http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0167100
[10] How can I use parallelism to make GATK tools run faster http://gatkforums.broadinstitute.org/gatk/discussion/1975/recommendations-for-parallelizing-gatk-tools
[11] Broad Institute (2016) Reference implementation: PairedEndSingleSampleWf pipeline. https://software.broadinstitute.org/gatk/documentation/article?id=7899
[12] Big Data and Analytics on IBM Power Systems https://www.ibm.com/developerworks/community/blogs/f0f3cd83-63c2-4744-9021-9ff31e7004a9/entry/Apache_Spark_Runs_2X_Faster_on_IBM_s_POWER8?lang=en

About the authors

Takeshi Ogasawara, PhD is a professor at INIAD Toyo University. He had been working on new techniques of compilers, runtime, and middleware at IBM Research, Tokyo for over 25 years. His current focus is on performance optimization of genomic data analysis pipeline.

Yinhe Cheng, PhD, is a senior technical consultant in IBM Life Science and NGS Solution team. Dr. Cheng has more than 12 years of experience in genomics, life science, and HPC solution enablement and computational biology research. She is not only an expert in performance tuning and system sizing for life science solutions, but also has published in leading journals and conferences in the area of bioinformatics and data mining.

Kathy Tzeng, PhD is the world-wide technical lead for life science and genomics solutions within the IBM Systems Group. Since joining IBM in 2001, Dr. Tzeng has been working with technical teams across IBM, industry partners, and open source communities on the enablement and performance optimization of life science applications on IBM solutions. She has published patents, IBM Redbooks®, technical papers, book chapters, and peer-reviewed journals.

IBM Systems
3039 Cornwallis Road
RTP, NC 27709

Produced in the United States of America