Challenges and opportunities of next-generation sequencing: a high performance computing perspective

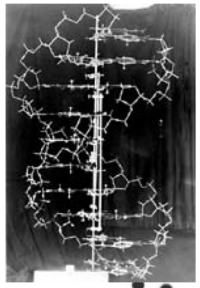
Korea Institute of Science and Technology Information (KISTI)
Div. of National Supercomputing

Hyojin Kang

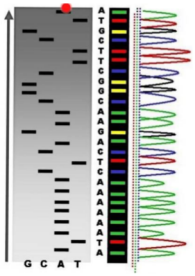
Contents

History of Human Genome Sequencing

1953
Discover the double helix structure of DNA



1977
Discover Sanger sequencing methods



1983
PCR method is invented

2003
Human Genome Project

2005
Development of NGS methods

2010
Sequencing of WGS cancer genomes

2012
Initiation of 100K Genome Project in UK

Genomics england

2016
"All of Us" 1M WGS by NIH

All of Us RESEARCH PROGRAM
The Precision Medicine Initiative

2018-2023
UK plans to sequence 5M WGS

“Countries in 100K Genome Club”

Genomics england

UK (100,000)
→ 5,000,000

NIH National Institutes of Health
All of Us Research Program

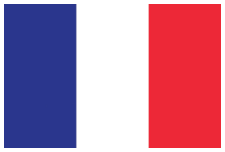
All of Us RESEARCH PROGRAM

US NIH (1,000,000)

FINNGEN

Finland, (500,000)

Japan, (100,000)



France, (235,000)

Australia, (100,000)

China, (100,000)

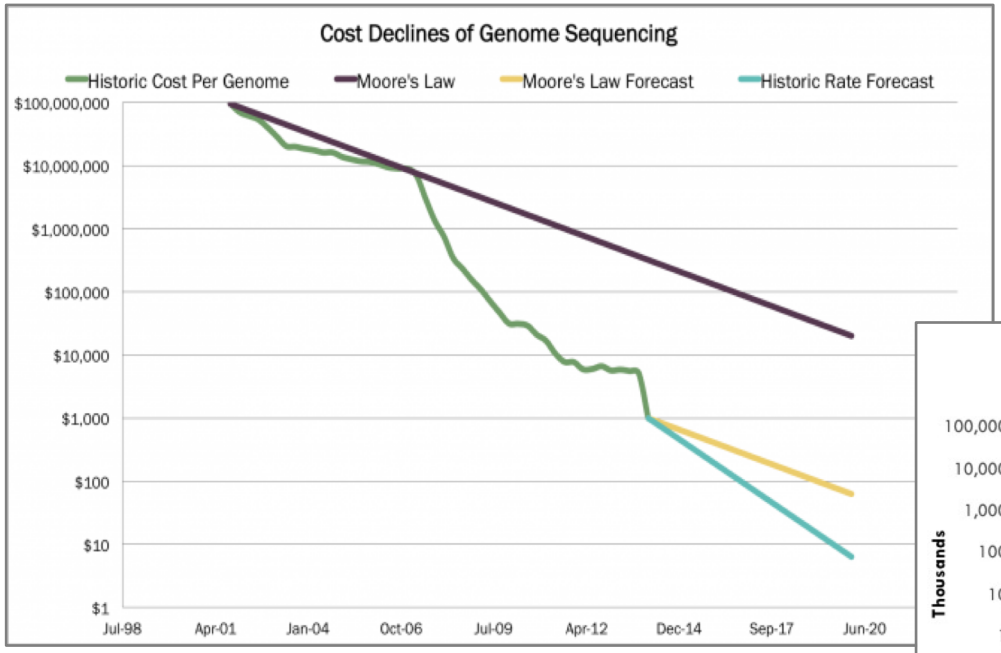
Saudi Arabia, (100,000)

https://www.clinicalomics.com/topics/biomarkers-topic/biobanking/10-countries-in-100k-genome-club/

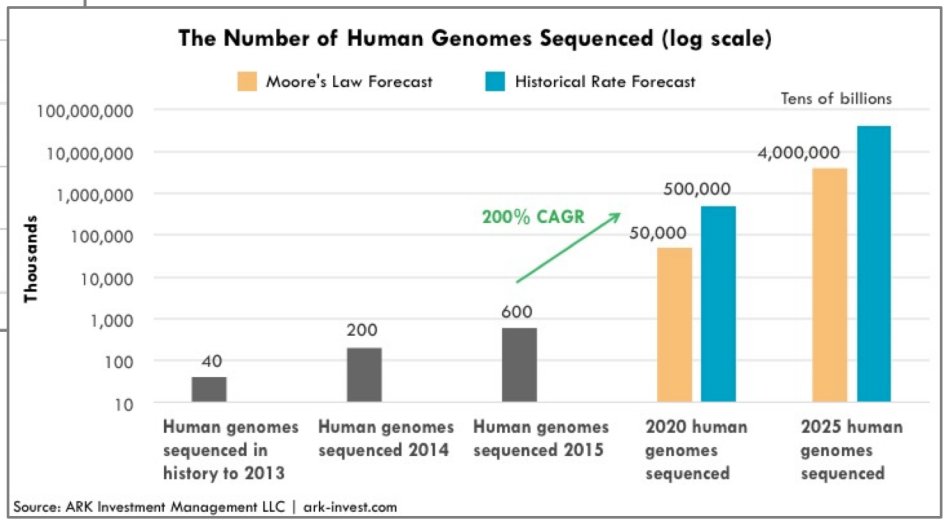
The cost drops faster than Moore's Law

- **Human Genome Project**
  - 1st human genome/10 years
- **NovaSeq, 2017 (Novogene)**
  - $100/30x coverage genome, 280,000 genomes/year

NovaSeq 6000 System
6Tb (~48 genomes) < 2 days



https://ark-invest.com/research/genome-sequencing

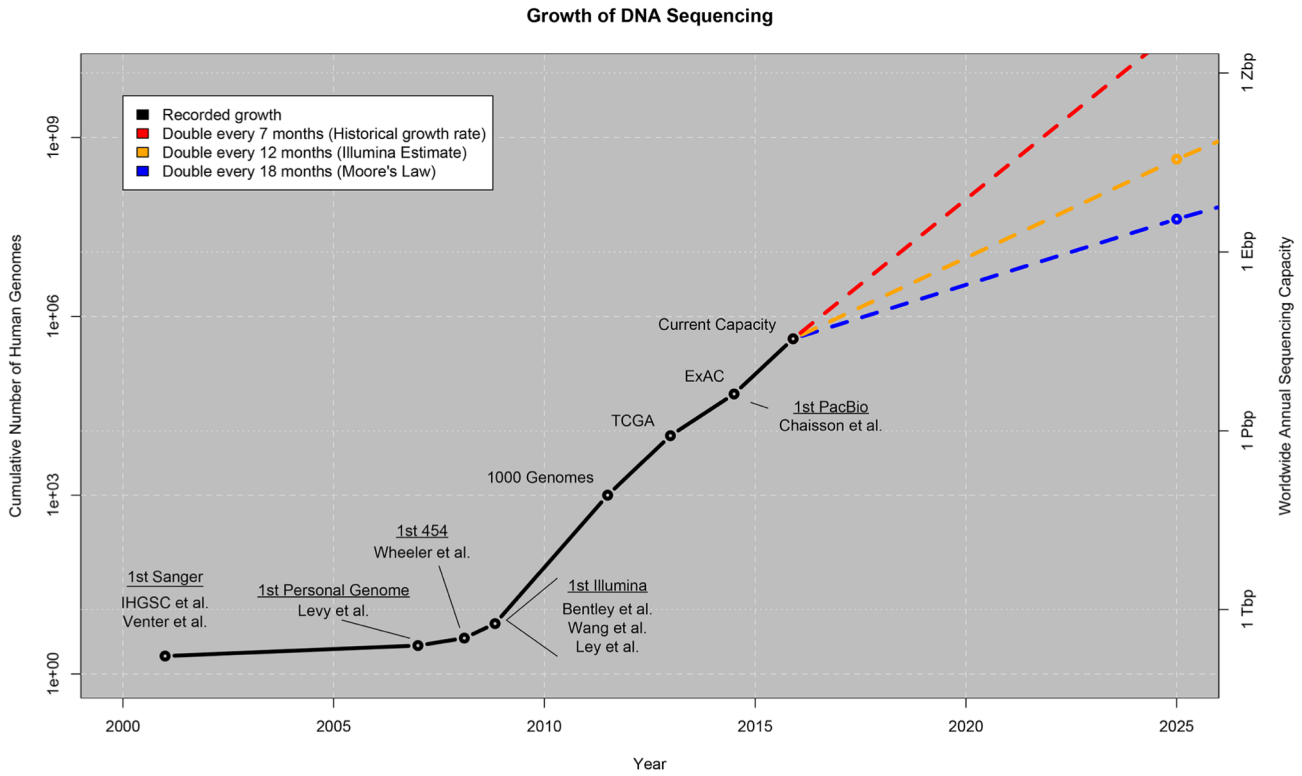


Growth of DNA sequencing data

Big Data: Astronomical or Genomical?

Zachary D. Stephens, Skylar Y. Lee, Faraz Faghri, Roy H. Campbell, Chengxiang Zhai, Miles J. Efron, Ravishankar Iyer, Michael C. Schatz, Saurabh Sinha, Gene E. Robinson

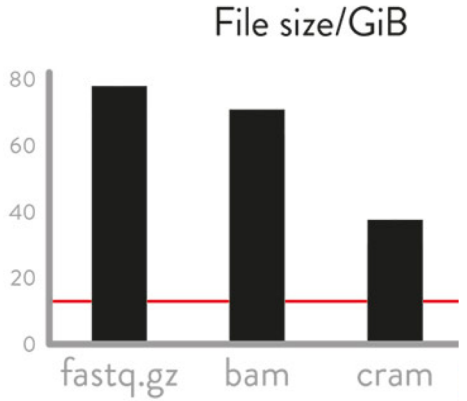
Published: July 7, 2015 • DOI: 10.1371/journal.pbio.1002195



2–40 exabytes of storage capacity will be needed **by 2025**

Computational Challenges

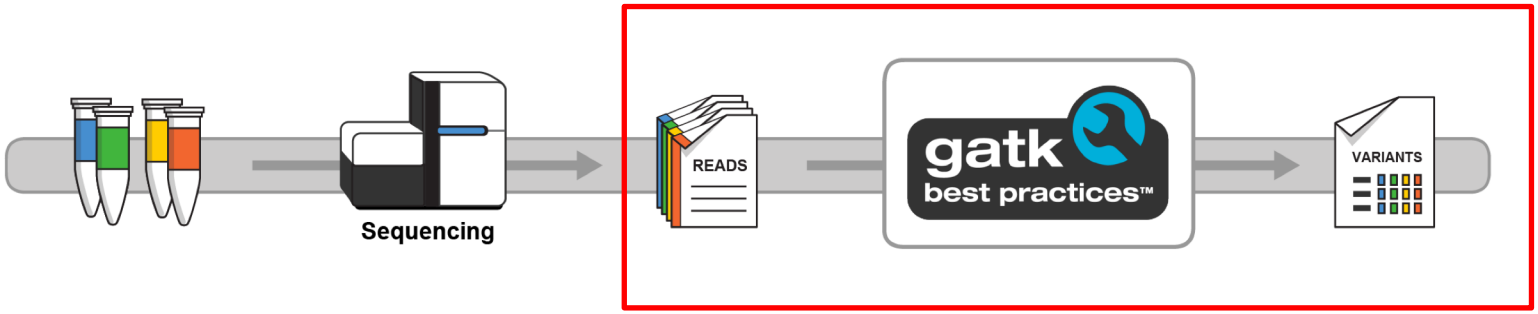
- **Huge Amount of Data**
  - 30x Hi-Seq WGS human: 80-90GB
  - 100K Genome Project: 180PB
  - Parallel I/O is critical

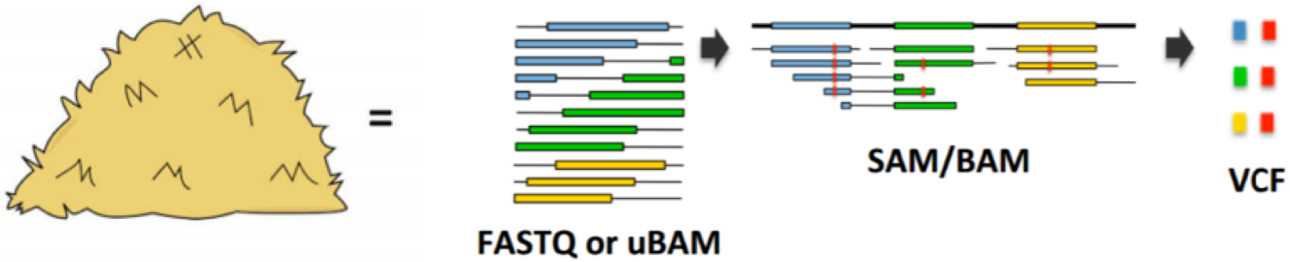


- **Extremely Long Running Time**
  - Due to large data size and high computation complexity
  - Long computation time: Genome data preprocessing + analysis pipeline: 4-5 days on a single server
  - Goal: several hours

- **Computation, storage, and I/O efficiency are critical**

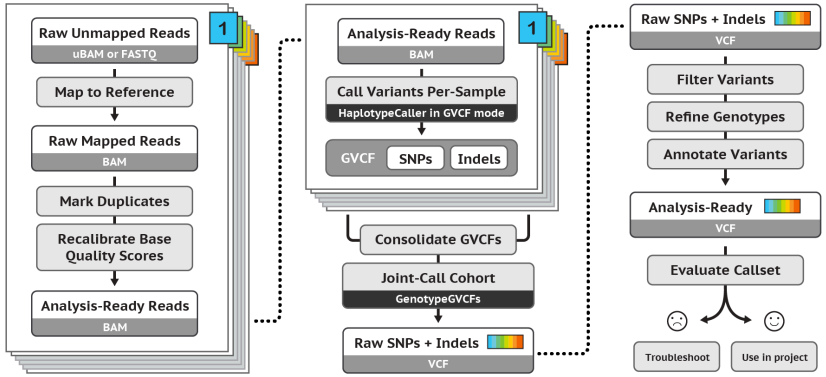
Genome analysis workflow





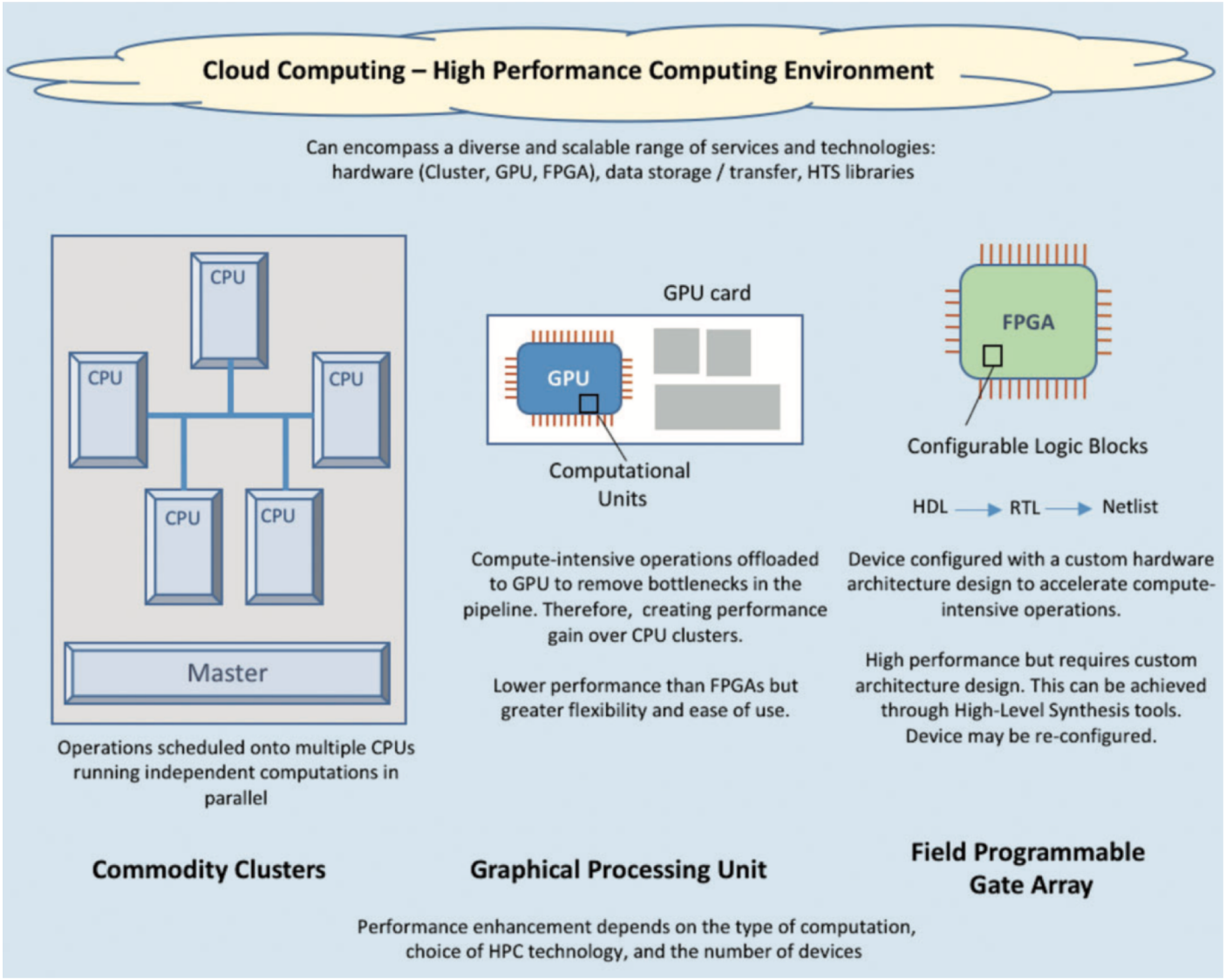
Genome Analysis Toolkit (GATK)

- Variant Discovery in High-Throughput Sequencing Data
- Developed by the Broad Institute



https://software.broadinstitute.org/gatk/

High Performance Computing Platforms



Lightbody, et. al., Briefings in Bioinformatics, 2018

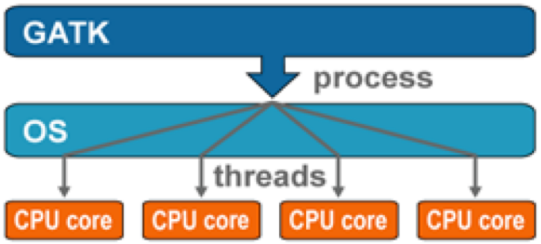
Optimization and acceleration

- **Commodity clusters**
  - Optimization 1: Multi-threading
  - Optimization 2: Scale-Out
  - Optimization 3: I/O Optimization
- **GPU, FPGA**
  - Optimization 4: Scale-Up using accelerators
- **Cloud computing**
  - Optimization 5: Resource-aware job scheduler

1) Multi-threading: by Intel

- **Single-Thread vs. Parallelized Run**

~6X speed-up

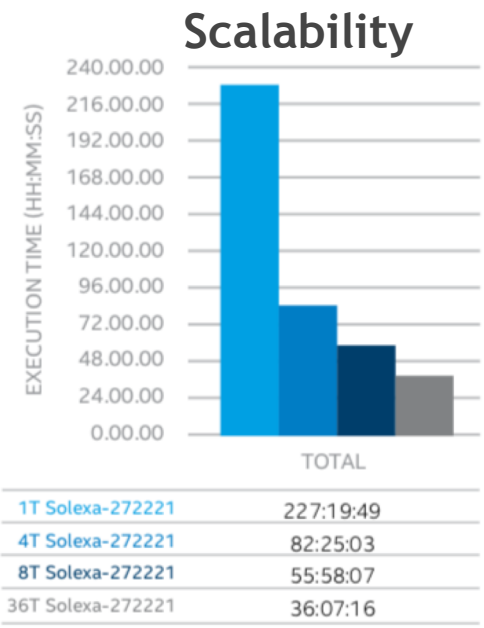


intel select solution

BROAD INSTITUTE

The two organizations have collaborated on computing infrastructure and software optimization for years

	2014	2016	Performance
Tools	Single-Thread Solexa-272221	36 Thread Solexa-272221	Speedup (x)
BWA Mem	92:03:19	3:50:58	23.9
Picard SortSam	7:55:51	6:36:35	1.2
Picard MarkDuplicates	6:34:47	5:45:15	1.1
GATK RealignerTargetCreator	5:58:20	0:18:56	18.9
GATK IndelRealigner	7:24:38	3:52:38	1.9
GATK BaseRecalibrator	19:49:07	1:56:07	10.2
GATK PrintReads	23:54:38	7:28:08	3.2
GATK HaplotypeCaller	63:39:09	6:18:39	10.1
Total Execution Time	227:19:49	36:07:16	6.3



https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/deploying-gatk-best-practices-paper.pdf

1) Multi-threading: GATK4

~5X speed-up

Speed improvements

Baseline benchmarking on ordinary hardware

- Av. 2x on ported GATK3 tools
- Up to 6x on re-implemented Picard tools

➔ Up to 5x speedup for full pipeline

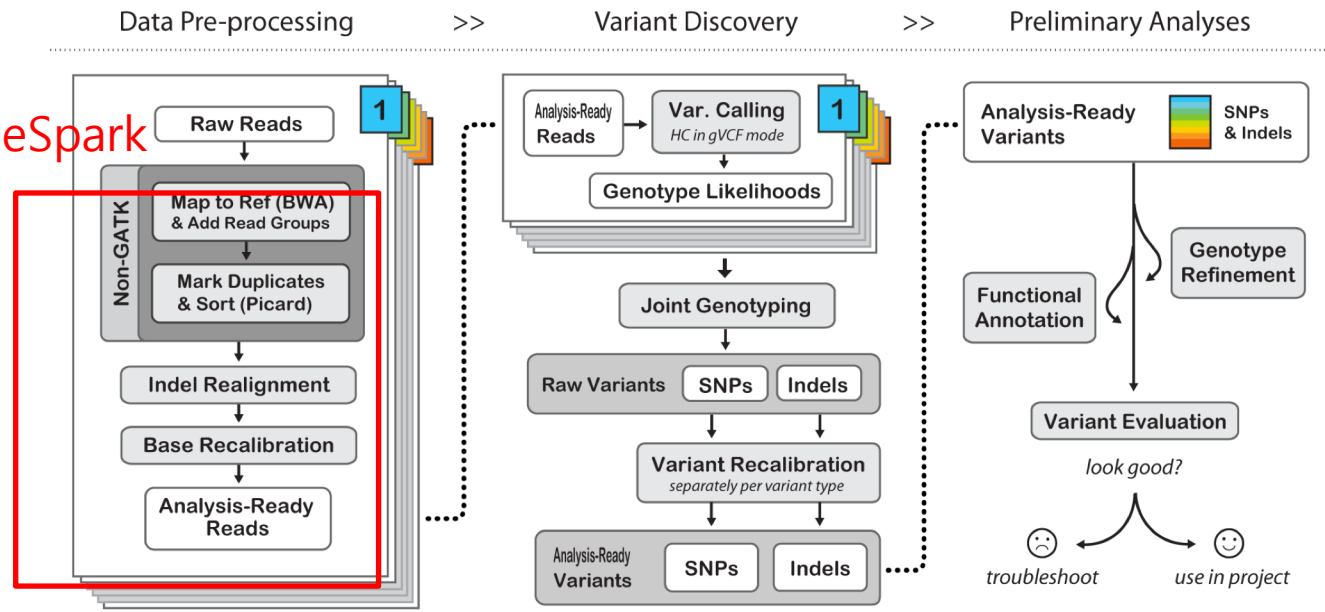
	Picard + GATK 3	GATK4
SortSam / Sort	24:26	3:52
MarkDuplicates	23:50	6:12
BaseRecalibrator	10:38	6:42
HaplotypeCaller	9:37	4:20

Additional pipeline speedups are available through Spark or scatter-gather parallelism

2) Scale-out: GATK4 Performance

- **GATK4 performance of whole genome analysis**
  - GATK 3.6: 46 hours on 12-core CPU (exclude variant calling)
  - GATK 4: 4.6 hours on 3 servers with a total of 108 cores (exclude variant calling)

GATK4: ReadsPipelineSpark

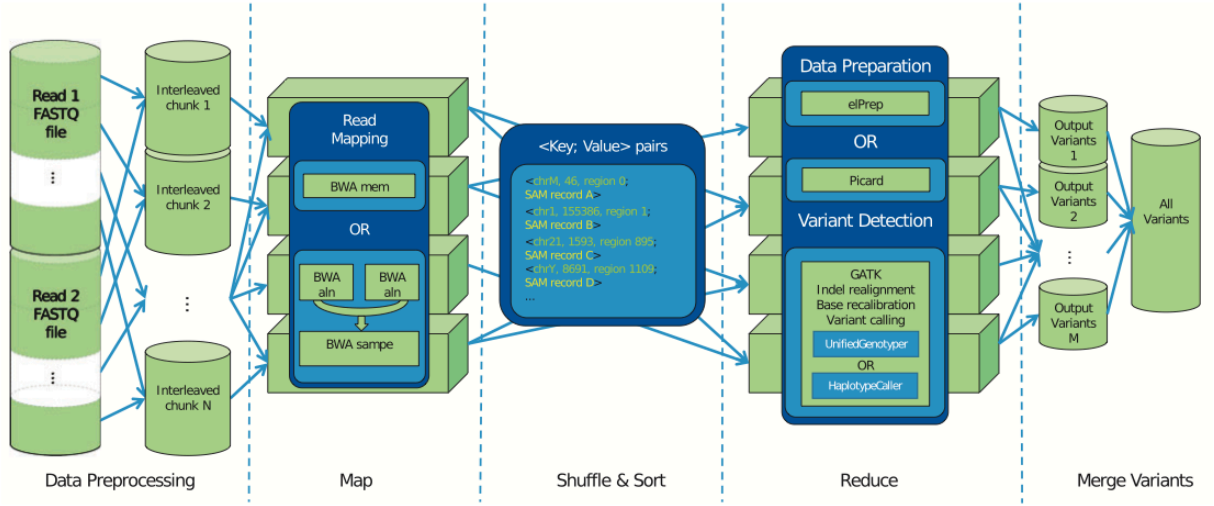


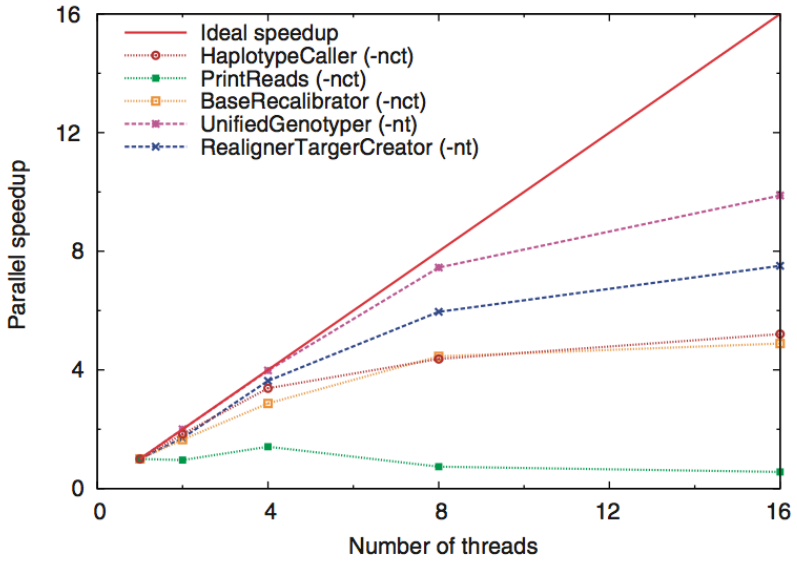
https://software.broadinstitute.org/gatk/documentation/article.php?id=9881

2) Scale-out: Halvade

- **Hadoop MapReduce based pipeline**

~18X speed-up

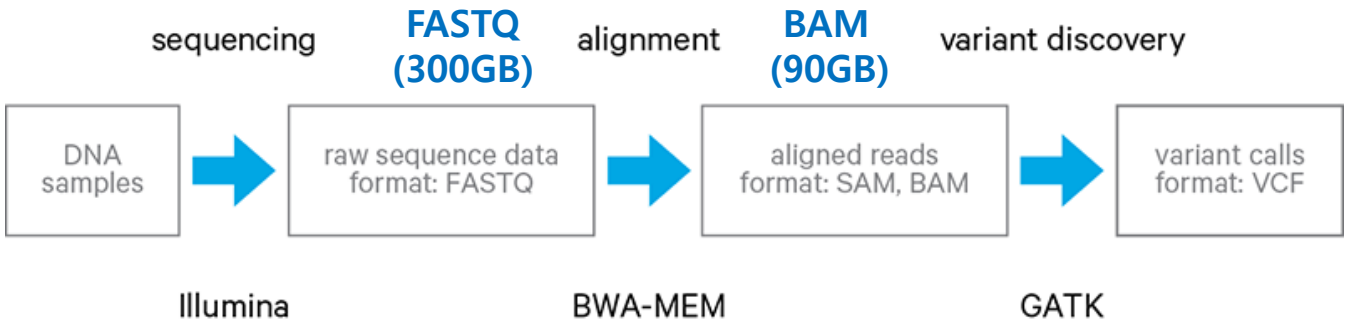




Cluster	No. worker nodes	No. parallel tasks	No. CPU cores	Runtime
Intel Big Data cluster	1	3	18	47 h 59 min
	4	15	90	9 h 54 min
	8	31	186	4 h 50 min
	15	59	354	2 h 39 min
Amazon EMR	1	4	32	38 h 38 min
	2	8	64	20 h 19 min
	4	16	128	10 h 20 min
	8	32	256	5 h 13 min
	16	64	512	2 h 44 min

Decap et. al, 2015, Bioinformatics

3) I/O Optimization

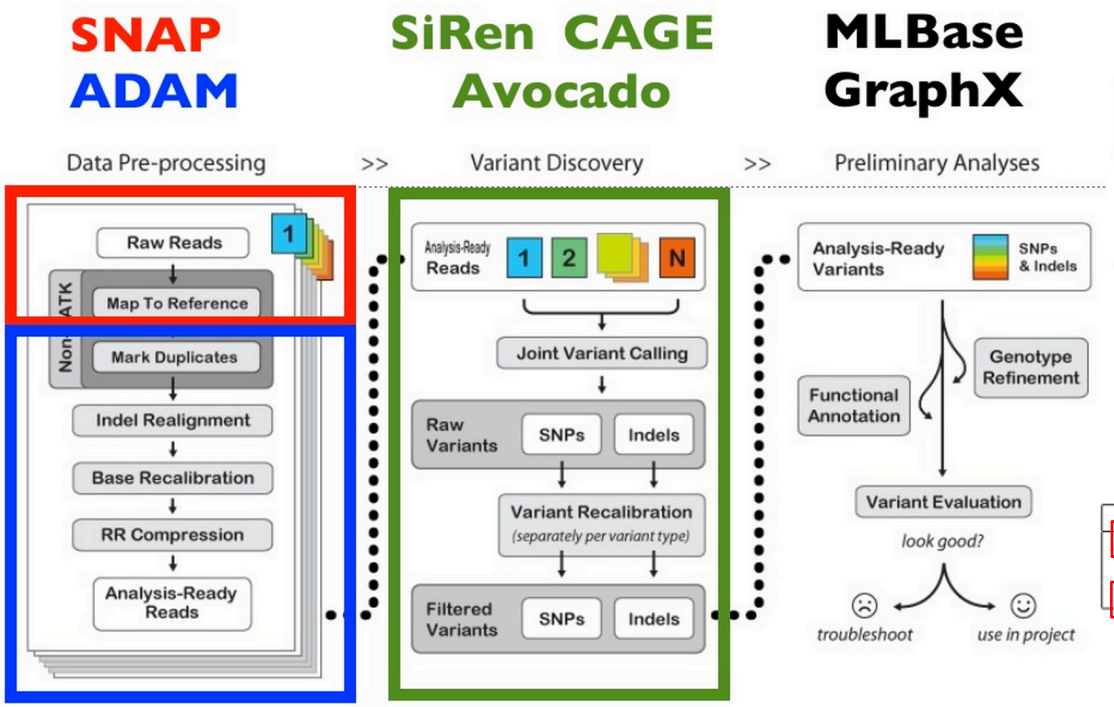


- **Genome analysis process continues to read / write repeatedly**
- **Optimizing or eliminating disk accesses can reduce time**
  - Replacing the HDDs with **SSD**s can bring up to ~4x speed-up
  - **Apache Spark**

https://blog.cloudera.com/blog/2016/04/genome-analysis-toolkit-now-using-apache-spark-for-data-processing/

3) I/O Optimization: ADAM

- ADAM (Avro Data Alignment Map) ~50X speed-up
  - The data are stored in **Parquet**, a columnar data storage using **Avro** serialized file formats
  - Reduce I/O load by providing **in-memory data access** for the **Spark pipeline**



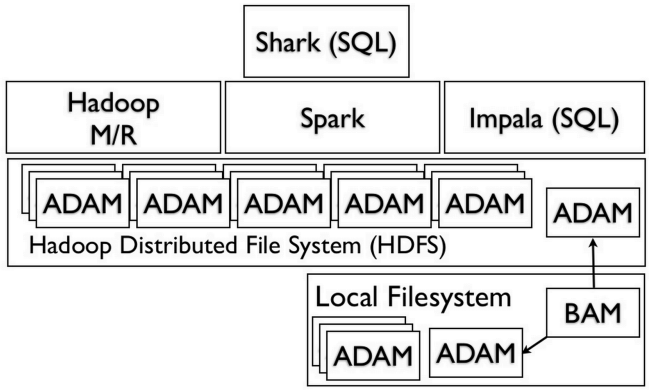


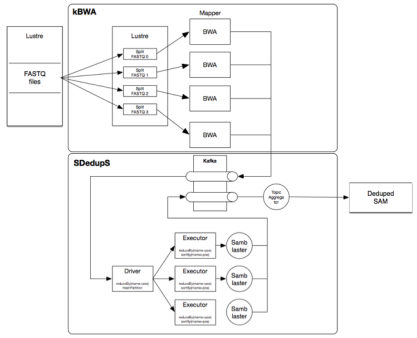
Table 10: Sort NA12878

Software	EC2 profile	Wall clock time
Picard Tools 1.103	1 hs1.8xlarge	17h 44m
ADAM 0.5.0	32 cr1.8xlarge	33m
ADAM 0.5.0	100 m2.4xlarge	21m

3) I/O Optimization: KISTI

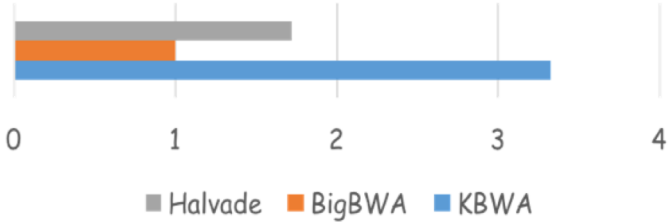
- **Hadoop on Lustre**
  - Omitting data stage-in/out process through utilization of Luster-based Hadoop

~4.7X speed-up

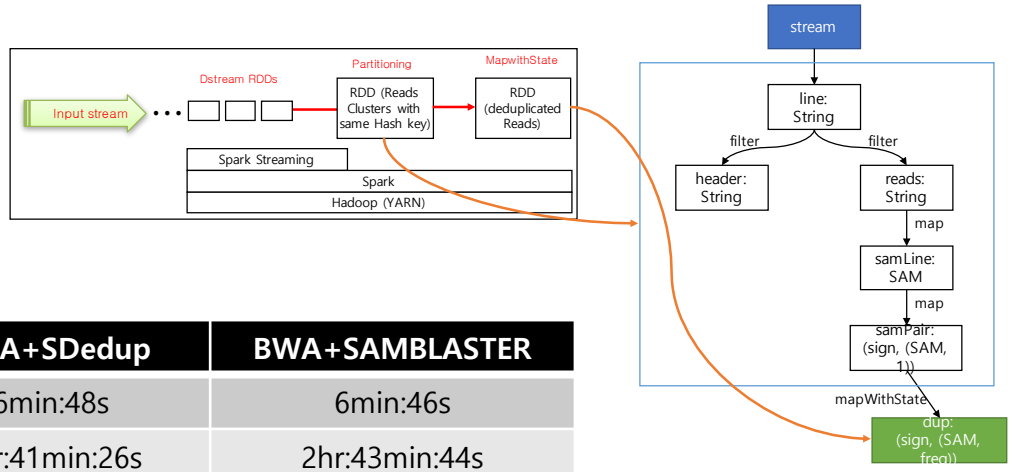


- **DeDup with Spark Stream**
  - Duplicate sequencing algorithm for large-capacity genome data through spark

190% faster than other Hadoop-based tools



Improving DeDup performance with Spark Stream

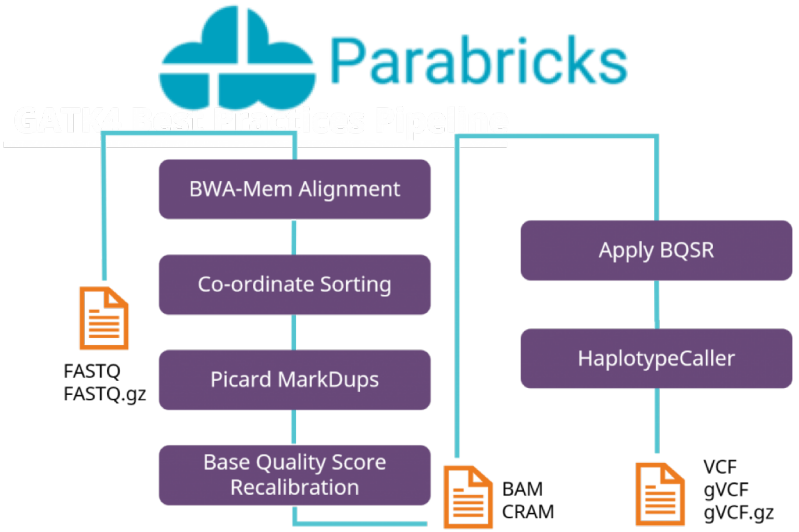


Data Size	# of Reads	BWA Only	BWA+SDedup	BWA+SAMBLASTER
712MB	2,813,850	6min:46s	6min:48s	6min:46s
18GB	24,777,542	2hr:39min:15s	2hr:41min:26s	2hr:43min:44s

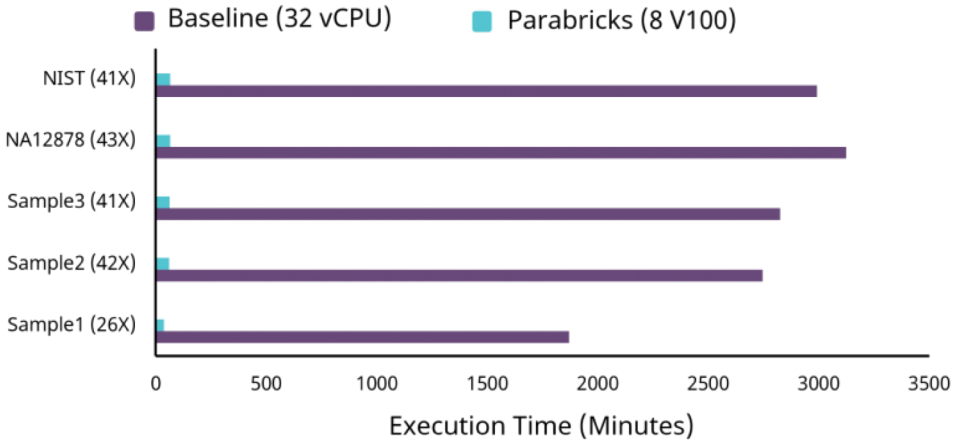
4) Acceleration with GPU: Parabricks

- GPU accelerated germline variant calling (SNVs and Indels) pipeline
- Uses the exact same algorithms as the BWA-GATK4
- Reduces the time 30 hours (BWA-GATK4) to 45 minutes

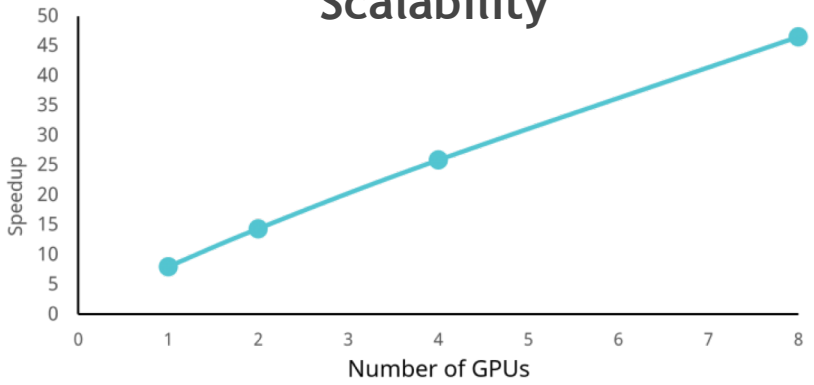
~40X speed-up



Performance

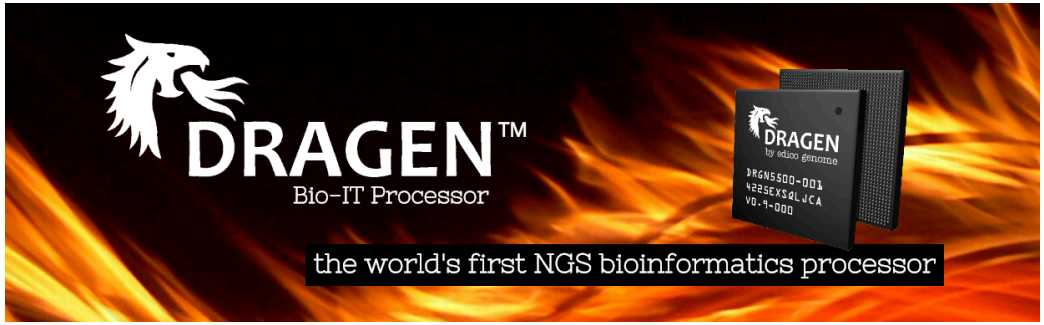


Scalability

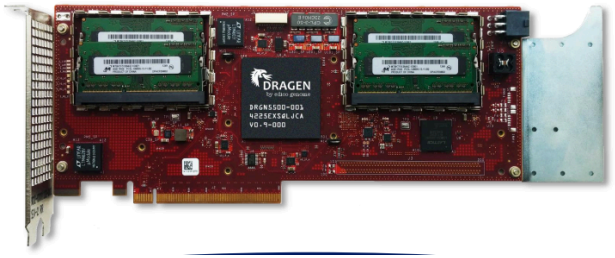


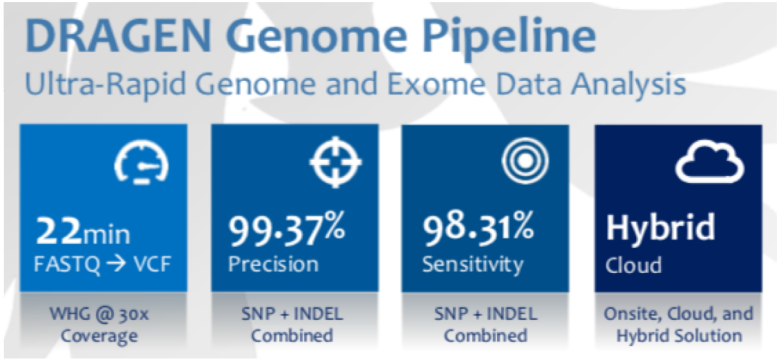
https://www.parabricks.com/germline/

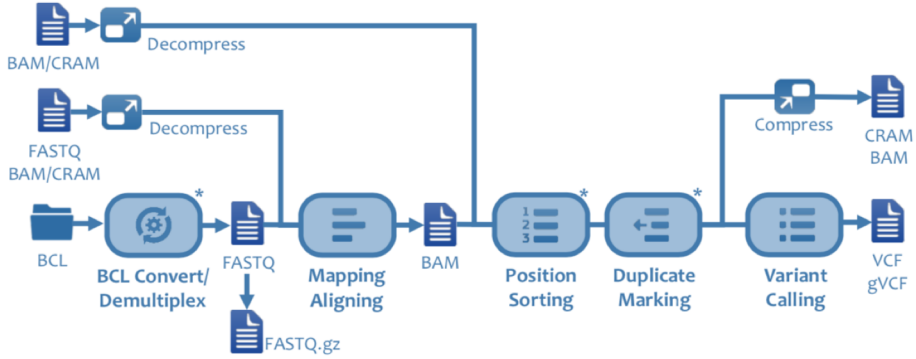
4) Acceleration with FPGA: DRAGEN



~82X speed-up







Dataset	Pipeline Configuration	DRAGEN	BWA + GATK-HC	DRAGEN Speed Up
SRA056922 NA12878 @ 30x	FASTQ to VCF	0:22:11	30:20:20	~82X
	FASTQ to gVCF	0:28:31	36:53:29	~78X
Garvan Lane1 NA12878 @ 37x	BCL to VCF	0:26:28	32:18:25	~73X
	BCL to gVCF	0:29:27	38:51:34	~79X

5) Resource-aware job scheduler

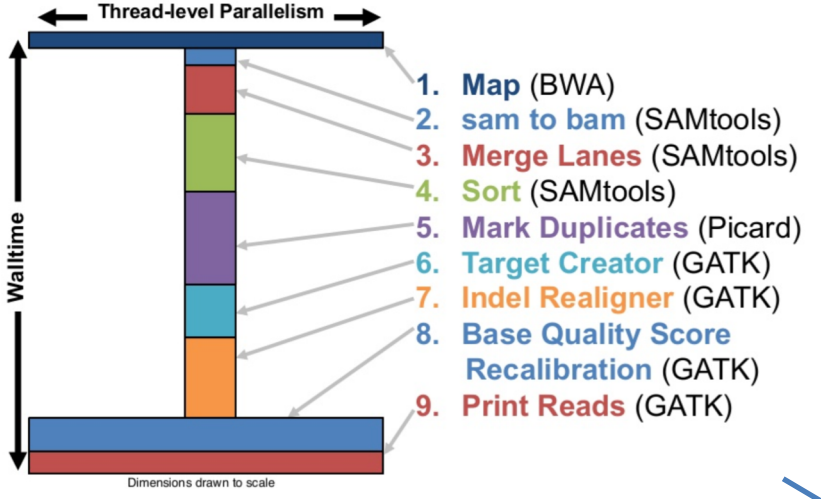
- **Different configuration options**
  - Runtime-optimal configurations for local clusters
  - Cost-optimal configurations in public cloud

	BWA-Mem	sam2bam (storage mode)	GATK BaseRecalibrator	GATK PrintReads	GATK HaplotypeCaller	GATK mergeVCF
CPU	Intensive. Close to 100% CPU utilization	~93% (initial phase) and ~40% in later phases	~70% CPU utilization	~70% CPU utilization	~40% CPU utilization	Less than 1% CPU utilization
Memory	Low memory consumption	Higher memory consumption with ~223 GB consumed	Total of 18 x Java threads with each thread customized with 10 GB ➔ 180 GB	Total of 18 x Java threads with each thread customized with 10 GB ➔ 180 GB	Not memory intensive	Not memory intensive
File data I/O access pattern	Pattern of writes followed by reads, Predominantly sequential I/O.	Write I/O predominantly sequential I/O. Read I/O is random access in units of 512 KiB	Predominantly read intensive. Read is mix of sequential and random I/O	Mix of read and write. Write I/O is mostly 512 KiB with mix of sequential and random. Read is mostly sequential	Mix of read and write. Write I/O is mix of sequential and random. Read is mostly sequential	Mix of read and write. Read and write I/O is predominantly sequential I/O.

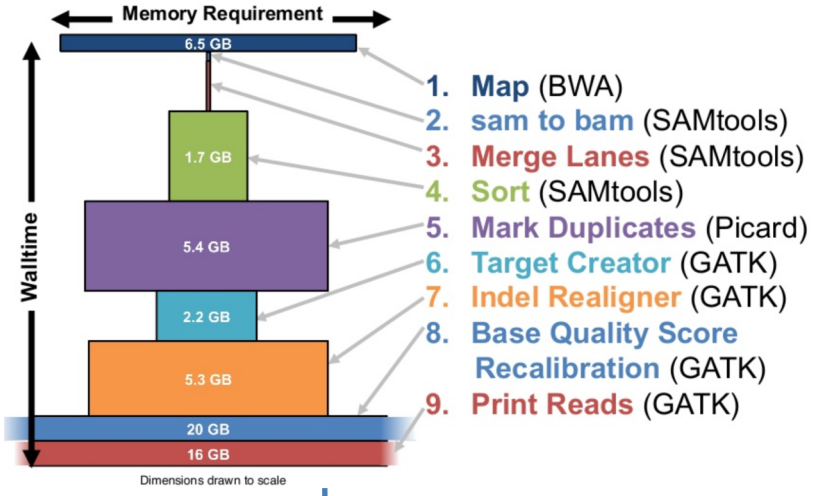
https://www.slideshare.net/UlfTroppens/ibm-spectrum-scale-best-practices-for-genomics-medicine-workloads

5) Resource-aware job scheduler

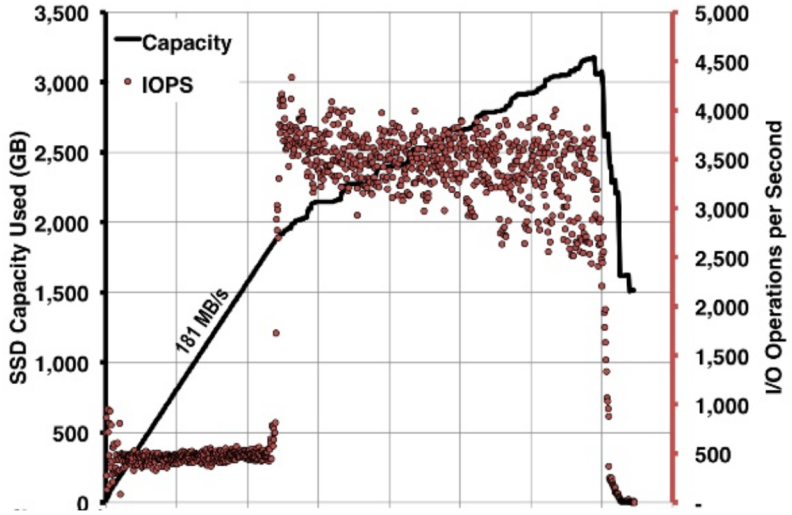
Multi-threading

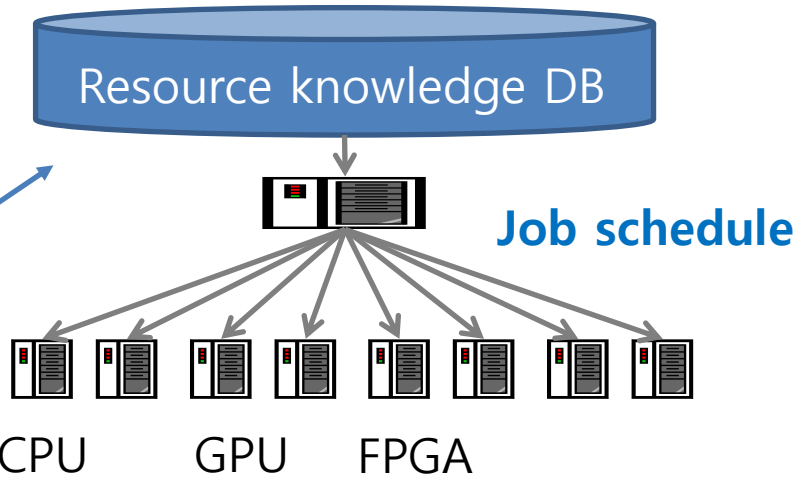


Memory requirement



Disk IO





https://www.slideshare.net/glennklockwood/janssen-presentation

Conclusion

- **Genome sequencing data will increase explosively in the future**
- **Reducing the computational burden for genome analysis is critical**
- **Efforts to optimize and accelerate are ongoing**
  - Optimization 1: Multi-threading: Intel, GATK4
  - Optimization 2: Scale-Out: GATK4
  - Optimization 3: I/O Optimization: HDFS, Spark
  - Optimization 4: GPU (Parabrick), FPGA (DRAGEN)
  - Optimization 5: Resource-aware job scheduler

References

- Jason Cong, (2017) Characterization and Acceleration for Genomic Sequencing and Analysis, IEEE International Symposium on Workload Characterization (IISWC)

- Yin, Z., Lan, H., Tan, G., Lu, M., Vasilakos, A. V., & Liu, W. (2017). Computing Platforms for Big Biological Data Analytics: Perspectives and Challenges. Computational and Structural Biotechnology Journal, 15(C), 403–411.

- Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., et al. (2015). Big Data: Astronomical or Genomical? *PLoS Biology*, *13*(7)

- Web sites
  - Illumina: https://www.illumina.com
  - Genome Analysis Toolkit: https://software.broadinstitute.org/gatk/
  - Halvade: https://github.com/biointec/halvade
  - ADAM: https://github.com/bigdatagenomics/adam
  - Parabricks: https://www.parabricks.com/
  - DRAGEN: https://www.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html

THANK YOU