The 47th IEEE International Symposium on Computer Architecture

Genesis: A Hardware Acceleration Framework for Genomic Data Analysis

Tae Jun Ham, David Bruns-Smith, Brendan Sweeney, Yejin Lee,
Seong Hoon Seo, U Gyeong Song, Young H. Oh,
Krste Asanovic, Jae W. Lee, Lisa Wu Wills

SEOUL NATIONAL UNIVERSITY

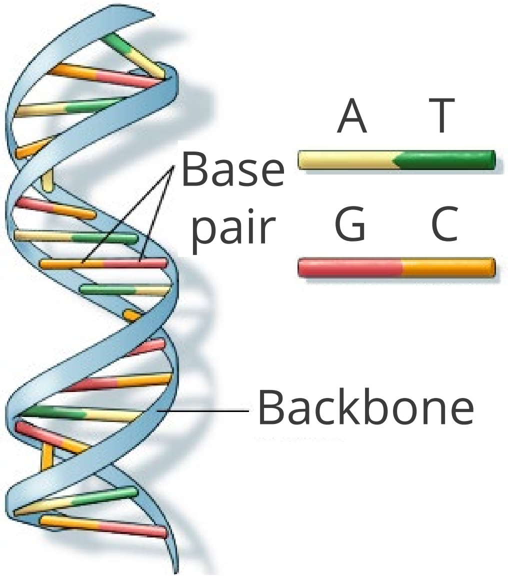
Berkeley
UNIVERSITY OF CALIFORNIA

SUNG KYUN KWAN UNIVERSITY(SKKU)

Duke
UNIVERSITY

Genomics and Genome Sequencing

- **DNA (deoxyribonucleic acid):** the chemical compound containing the instructions an organism needs to develop, live, and reproduce.
  - DNA is made of two paired strands, where each strand pair is represented with a single character (A, C, G, or T) that corresponds to the nucleotide base of a single pair
- **DNA sequencing (genome sequencing):** a process of identifying the base pair sequence for a DNA
- **Why is it important?**
  - Can identify if a person is susceptible to a specific disease
  - Can identify the type/variant of the cancer
  - Can be used for genetics research
  - Also used for COVID-19 researches (e.g., identification of the virus, virus variant analysis)

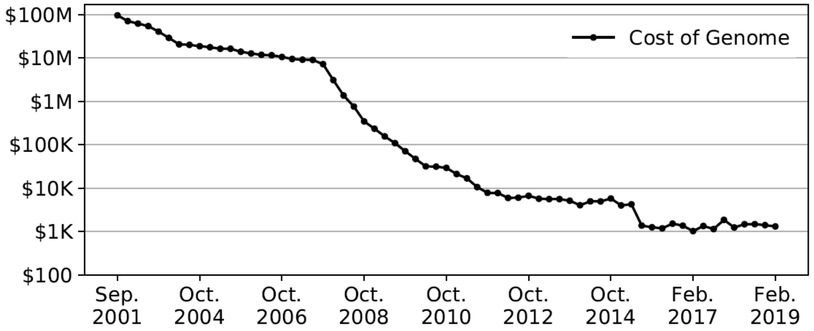


DNA

Source: U.S National Library of Medicine

Genomics and Genome Sequencing

- Genome Sequencing was very expensive, and time-consuming.
  - Human Genome Project cost $2.7B billion and took 13 years.
- Next-Generation Sequencing (NGS) technology enabled the rapid sequencing of a whole genome
  - Whole genome sequencing now costs $300-$700[1] and takes less than an hour per genome[2]
- Genome sequencing comes with a huge computational demand
  - Data obtained from Genome sequencing instruments (i.e., raw reads) needs to be processed with the various algorithms
  - This process is called Secondary Analysis

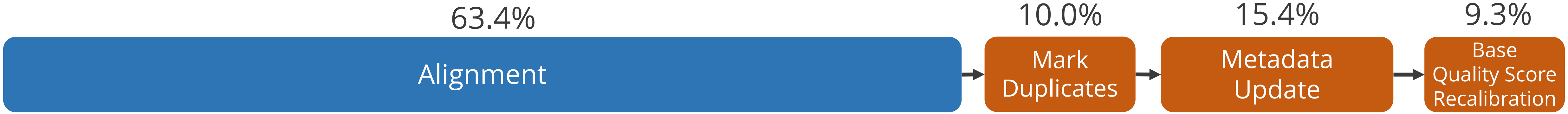


Cost of Genome Sequencing

Source: U.S National Human Genome Institute

[1] https://nebula.org/whole-genome-sequencing/
[2] https://sapac.illumina.com/systems/sequencing-platforms/novaseq/specifications.html

Advent of Hardware Accelerators for Genome Sequencing



63.4% Alignment → 10.0% Mark Duplicates → 15.4% Metadata Update → 9.3% Base Quality Score Recalibration

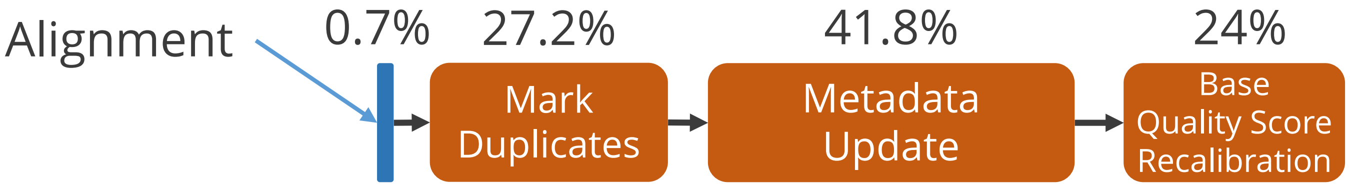
GATK4 Best Practices Data Preprocessing Pipeline Runtime Breakdown (measured on Intel Xeon 8-cores)

(Miscellaneous stages accounting for 1.9% of the runtime are omitted)

- Complex stage such as Alignment takes most of the runtime and thus has been targets for many hardware accelerators
  - GenAx [ISCA '18], Darwin [ASPLOS' 18], Guo et al. [FCCM '19]
  - Other complex stages such as Variant Calling (downstream) are accelerated as well

- Advent of hardware accelerators shifts the bottleneck to simple data-manipulation operations

Advent of Hardware Accelerators for Genome Sequencing



GATK4 Best Practices Data Preprocessing Pipeline Runtime Breakdown (measured on Intel Xeon 8-cores)

(Miscellaneous stages accounting for 1.9% of the runtime are omitted)

- Complex stage such as Alignment takes most of the runtime and thus has been targets for many hardware accelerators
  - GenAx *[Fujiki et al., ISCA '18]*, Darwin *[Turakhia et al., ASPLOS' 18]*, *[Guo et al., FCCM '19]*
  - Other complex stages such as Variant Calling (downstream) are accelerated as well

- Advent of hardware accelerators shifts the bottleneck to simple data-manipulation operations
  - Assuming GenAx throughput (4058K reads/s), the alignment only takes 0.7% of the total data preprocessing runtime
  - Data-manipulation operations accounts for 93% of the total runtime

Genesis: A Hardware Acceleration Framework for Genomic Data Analysis

Genesis is a framework that enables the users to easily design a cloud-deployable hardware accelerator for the genomic data-manipulation operations

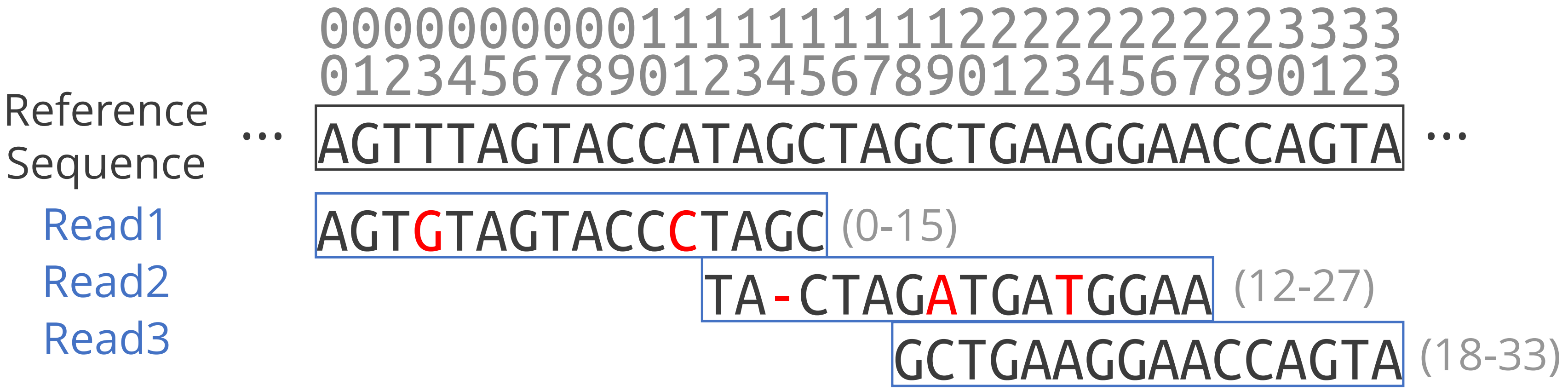
1. A user utilizes **Genesis SQL Frontend** to represent the target data-manipulation operation in a way that can be easily mapped to the hardware
2. Components in **Genesis Hardware Library** (configurable accelerator building blocks) is used to construct a dataflow pipeline for the specified SQL query
3. **Genesis Backend** automatically augments the pipeline with parallelism, deploys it on cloud FPGA, and allows a user to access it with high-level API

Presentation Outline

- Genomics and Genome Sequencing
- Genesis: A Hardware Acceleration Framework for Genomic Data Analysis
  - Genesis SQL Frontend
  - Genesis Hardware Library
  - Genesis Backend
  - Genesis-generated HW accelerators
- Evaluation
- Conclusions

Genesis SQL Interface

- **Observation**: Most simple data manipulation operations for genomic data can be easily represented with a SQL Query[1,2] on genomic data represented in tabular form
- **Key Data Types**: Reference and Reads
  - Reference: A reference genome sequence for an individual organism of a species (e.g., human)
  - (Aligned) Reads: A fragment of the genome sequence measured using sequencing instruments with some metadata



```
                       0000000000111111111122222222223333
                       0123456789012345678901234567890123
Reference Sequence ... AGTTTAGTACCATAGCTAGCTGAAGGAACCAGTA ...
Read1                  AGTGTAGTACCCTAGC (0-15)
Read2                              TA-CTAGATGATGGAA (12-27)
Read3                                    GCTGAAGGAACCAGTA (18-33)
```

[1] Massie et al., ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing, UC Berkeley Tech Report, 2013

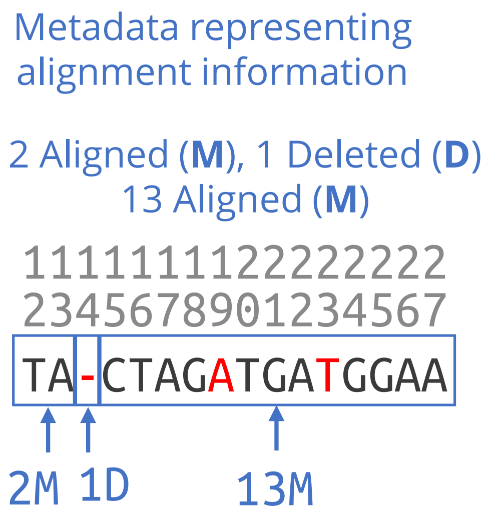
[2] Kozanitis et al., GenAp: a distributed SQL interface for genomic data, BMC informatics, 2016

Genesis SQL Interface (Tabular Data Representation)

- **Observation:** Most simple data manipulation operations for genomic data can be easily represented with a SQL Query[1,2] on genomic data represented in tabular form
- **Key Data Types**: References and Reads

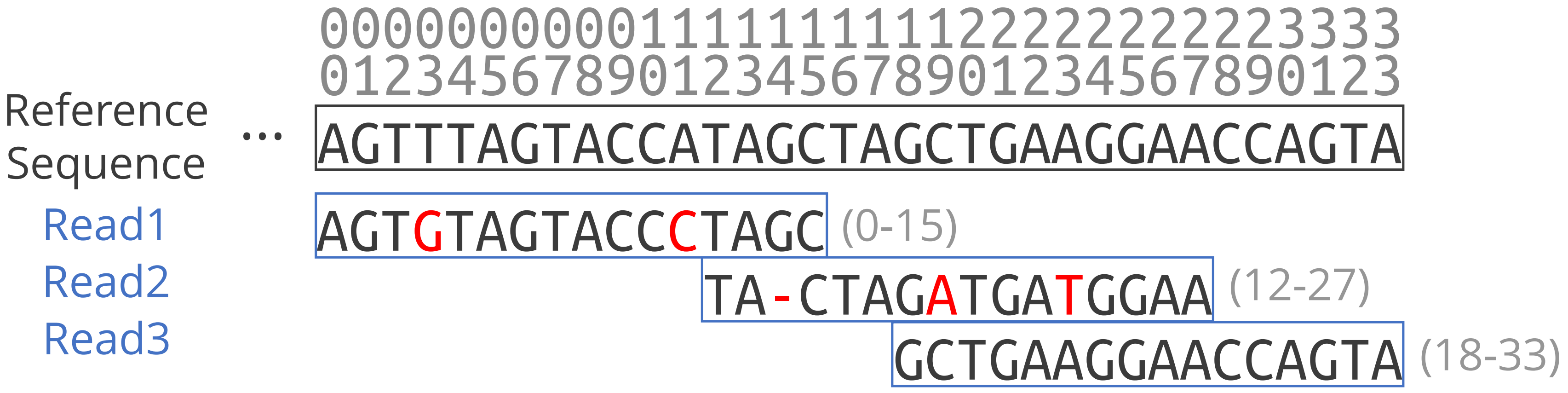
Reference Table	
POS	SEQ
0	AGTTTAGTACCATAGCTAG CTGAAGGAACCAGTA

(Simplified) Reads Table		
POS	SEQ	CIGAR
0	AGTGTAGTACCCTAGC	16**M**
12	TACTAGATGATGGAA	2**M**, 1**D**, 13**M**
18	GCTGAAGGAACCAGTA	16**M**



Metadata representing alignment information

2 Aligned (**M**), 1 Deleted (**D**)
13 Aligned (**M**)



[1] Massie et al., ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing, UC Berkeley Tech Report

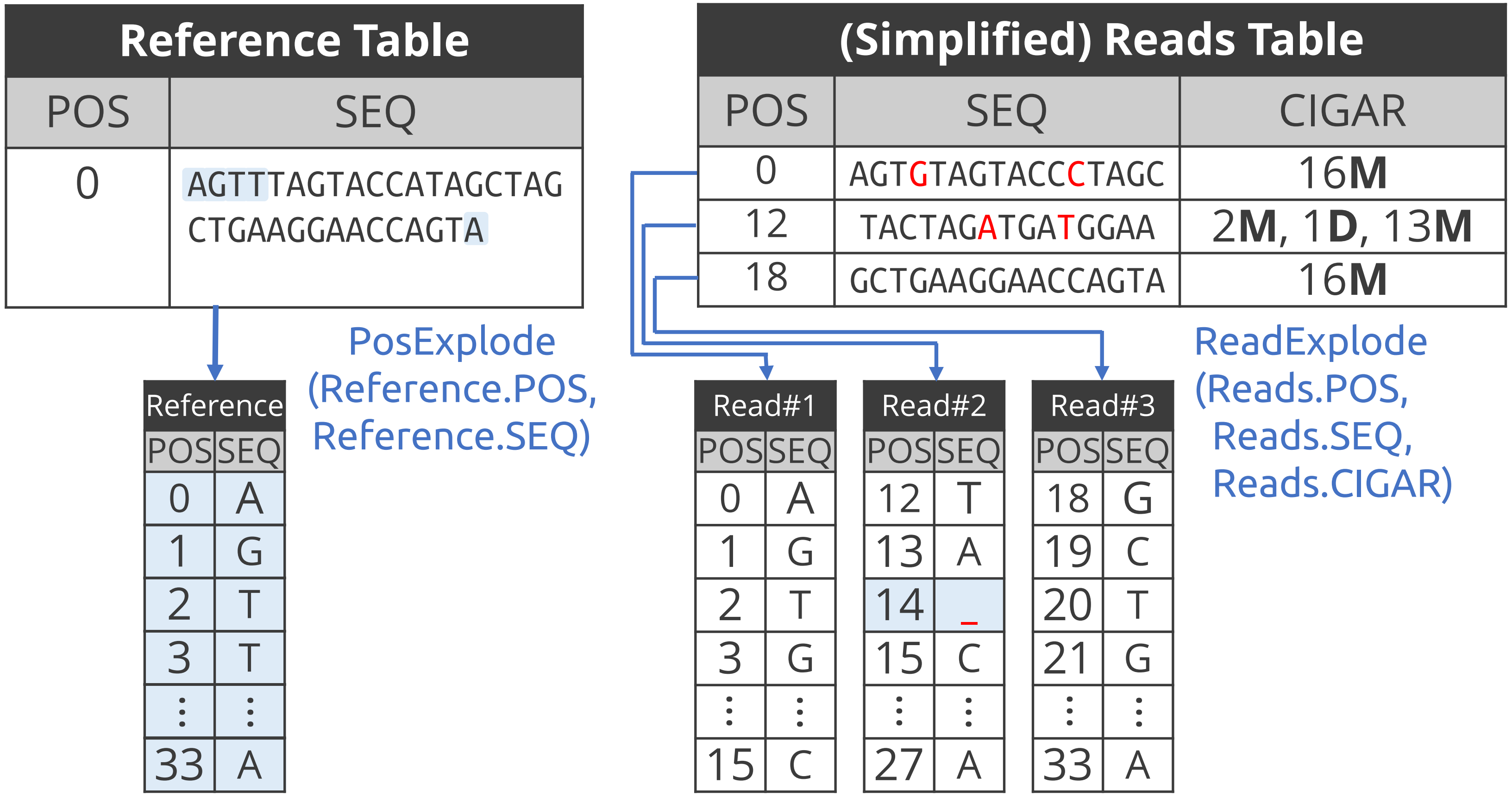
[2] Kozanitis et al., GenAp: a distributed SQL interface for genomic data, BMC informatics, 2016

Genesis SQL Interface (Operations)

- **(Common) Supported SQL Operations**:

  Select, **Where**, **GroupBy**, **Join**, **Limit** (i.e., select a subset of rows), **Count**, **Sum**, etc.

- **Additional Supported Operations**: PosExplode & ReadExplode



Reference Table	
POS	SEQ
0	AGTTTAGTACCATAGCTAG CTGAAGGAACCAGTA

(Simplified) Reads Table		
POS	SEQ	CIGAR
0	AGTGTAGTACCCTAGC	16**M**
12	TACTAGATGATGGAA	2**M**, 1**D**, 13**M**
18	GCTGAAGGAACCAGTA	16**M**

PosExplode (Reference.POS, Reference.SEQ)

Reference	
POS	SEQ
0	A
1	G
2	T
3	T
⋮	⋮
33	A

ReadExplode (Reads.POS, Reads.SEQ, Reads.CIGAR)

Read#1	
POS	SEQ
0	A
1	G
2	T
3	G
⋮	⋮
15	C

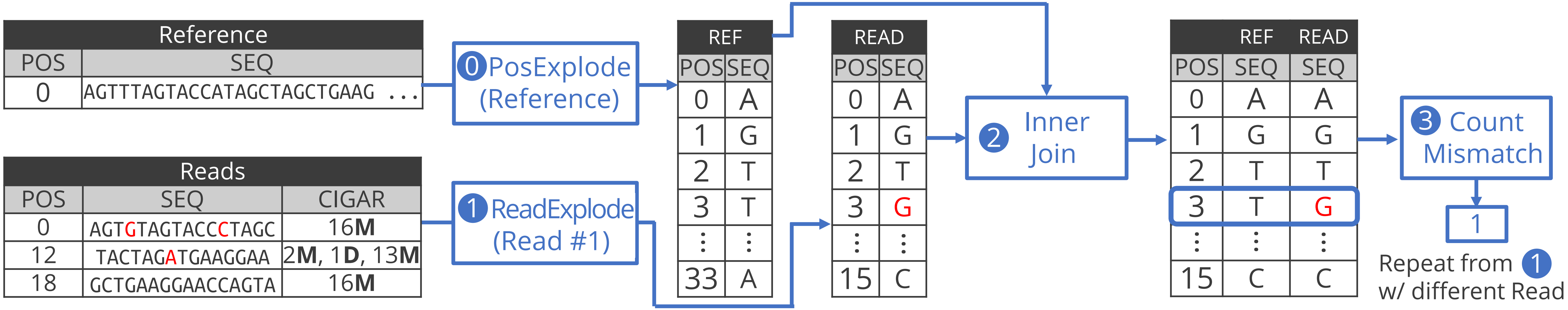
Read#2	
POS	SEQ
12	T
13	A
14	_
15	C
⋮	⋮
27	A

Read#3	
POS	SEQ
18	G
19	C
20	T
21	G
⋮	⋮
33	A

Genesis SQL Interface (Example App.)

Example Application

Compute the number of base pair mismatches between the reference and each read



Reference	
POS	SEQ
0	AGTTTAGTACCATAGCTAGCTGAAG ...

Reads		
POS	SEQ	CIGAR
0	AGTGTAGTACCCTAGC	16M
12	TACTAGATGAAGGAA	2M, 1D, 13M
18	GCTGAAGGAACCAGTA	16M

REF	
POS	SEQ
0	A
1	G
2	T
3	T
⋮	⋮
33	A

READ	
POS	SEQ
0	A
1	G
2	T
3	G
⋮	⋮
15	C

	REF	READ
POS	SEQ	SEQ
0	A	A
1	G	G
2	T	T
3	T	G
⋮	⋮	⋮
15	C	C

Repeat from 1 w/ different Read

```
CREATE TABLE REF AS
PosExplode (Reference.SEQ, Reference.POS)
FROM Reference
```

```
FOR R IN Reads:
  /* Step 1 */
  /* Step 2 */
  /* Step 3 */
END LOOP;
```

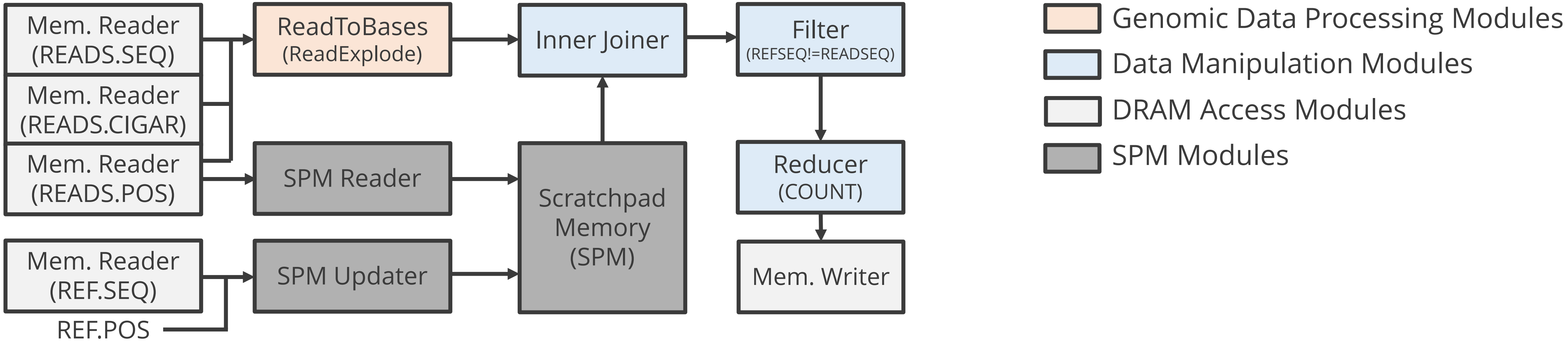
```
CREATE TABLE READ AS                         Step #1
ReadExplode (R.POS, R.SEQ, R.CIGAR) FROM R
```

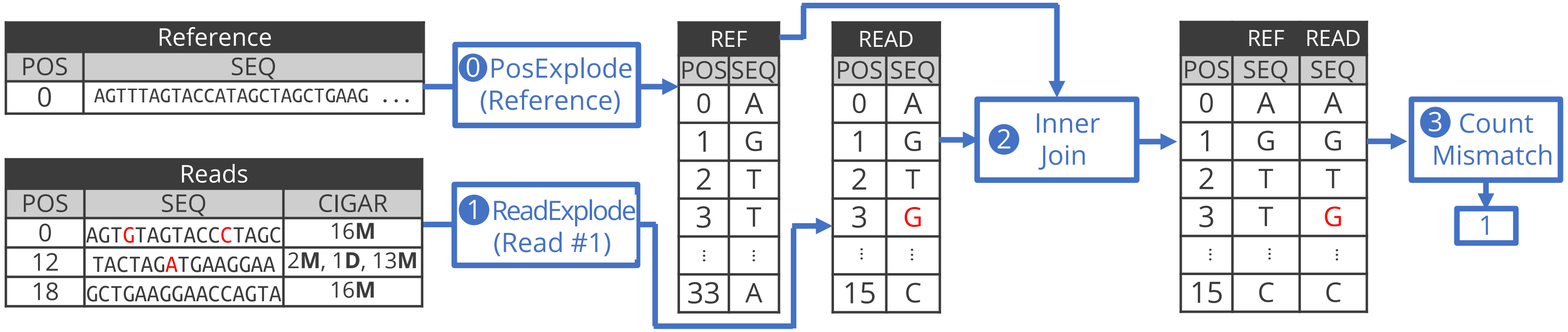
```
CREATE TABLE RefRead AS                      Step #2
SELECT READ.SEQ, REF.SEQ FROM READ
INNER JOIN (SELECT * FROM REF LIMIT 0, 15)
ON READ.POS = REF.POS INSERT INTO Output
```

```
SELECT SUM(READ.SEQ == REF.SEQ)              Step #3
FROM RefRead
```

Hardware Pipeline Design with Genesis HW Library

- SQL and its tabular data types map very well to the stream dataflow architecture
  - Q100 [ASPLOS’14], LINQits [ISCA ‘13], SDA [HotChips ’16]





Reference	
POS	SEQ
0	AGTTTAGTACCATAGCTAGCTGAAG ...

Reads		
POS	SEQ	CIGAR
0	AGTGTAGTACCCTAGC	16**M**
12	TACTAGATGAAGGAA	2**M**, 1**D**, 13**M**
18	GCTGAAGGAACCAGTA	16**M**

REF	
POS	SEQ
0	A
1	G
2	T
3	T
⋮	⋮
33	A

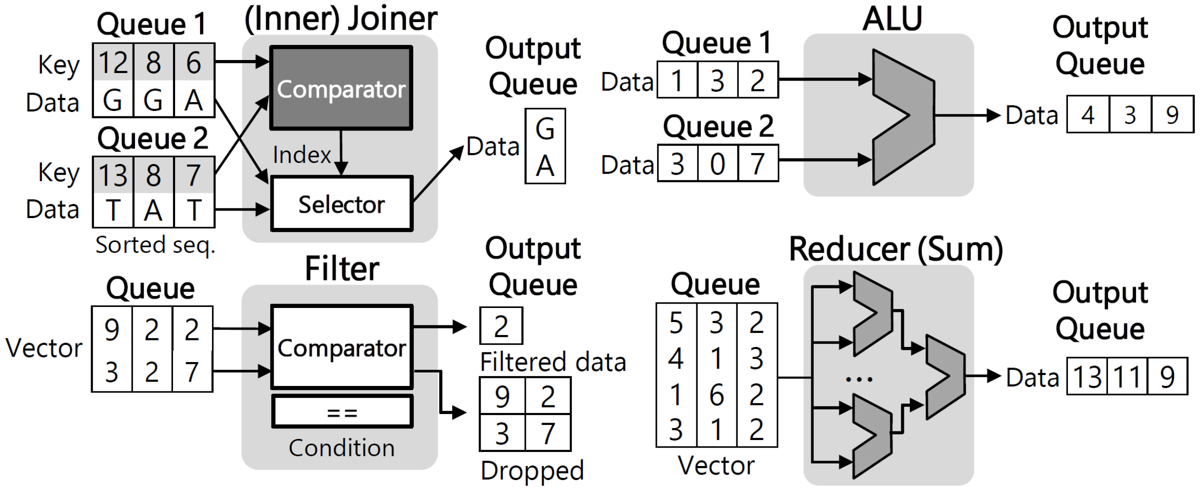
READ	
POS	SEQ
0	A
1	G
2	T
3	G
⋮	⋮
15	C

	REF	READ
POS	SEQ	SEQ
0	A	A
1	G	G
2	T	T
3	T	G
⋮	⋮	⋮
15	C	C

Compute the number of base pair mismatches between the reference and reads

Genesis HW Library

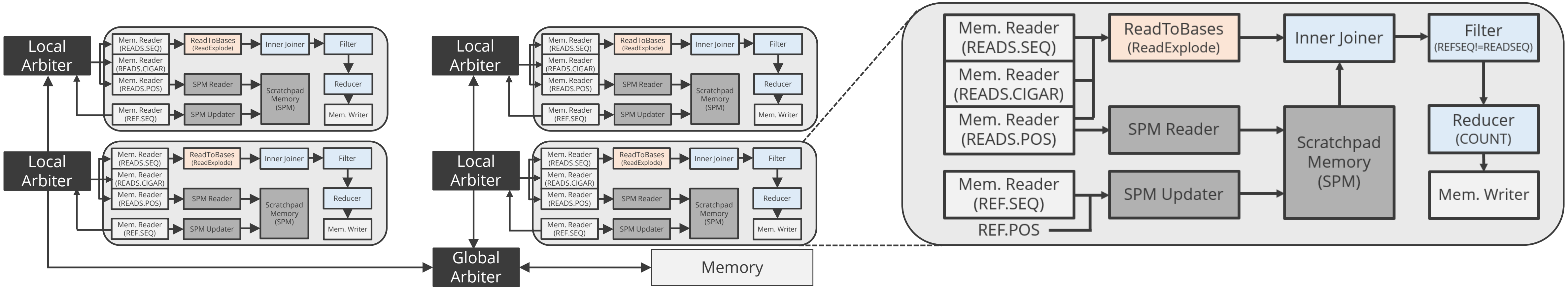
- **Genesis HW Library** includes four types of hardware modules
  - **Data Manipulation** modules
    - Joiner, Filter, Reducer, ALU
  - **Genomic Data Processing** modules
    - ReadToBases
  - **DRAM/SPM Access** modules
    - Sequential Read/Write with Prefetch & Buffered Write (DRAM)
    - Random/Sequential Read/Write & Atomic RMW (SPM)
  - **Custom** modules
    - User can integrate a custom simple computation module



Genesis Data Manipulation modules

Genesis Backend

- **Genesis Backend** automatically exploits the abundant parallelism within the genomics data manipulation operations

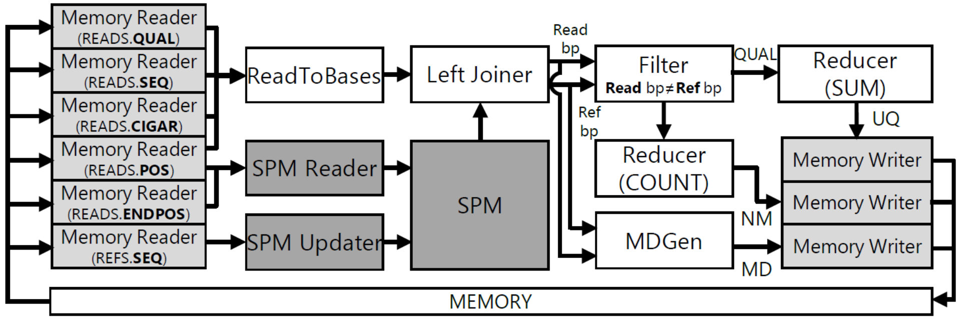


- Also provides high-level user-API that enables users to easily control the accelerator data movements and computation
  - `void configure_mem (void* addr, int elemsize, int len, string colname, int pipelineID)`
  - `void run_genesis(int pipelineID)`
  - `bool check_genesis(int pipelineID)` / `void wait_genesis(int pipelineID)`
  - `void flush_genesis(int pipelineID)`

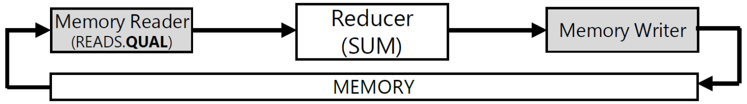
Genesis-Generated Hardware Accelerators

- Genesis framework is used to accelerate three data-manipulation operations in Data Preprocessing Phase of the GATK4 Genome Sequencing Pipeline
  - (Portion of) Mark Duplicates, Metadata Update, BQSR (Covariate Table Construction)
  - Accounts for more than 80% of the data preprocessing phase once the alignment stage is hardware-accelerated with GenAx[1]

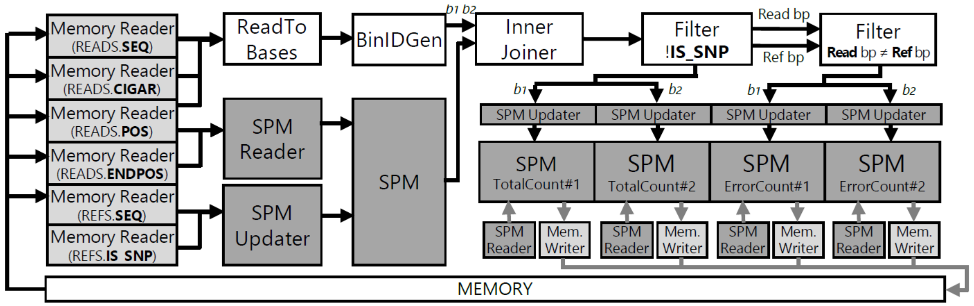
Metadata Update



Mark Duplicates (Quality Score Reduction)



BQSR (Covariate Table Construction)



[1] D. Fujiki, A. Subramaniyan, T. Zhang, Y. Zeng, R. Das, D. Blaauw, and S. Narayanasamy, "GenAX: a genome sequencing accelerator," in ISCA 2018

Presentation Outline

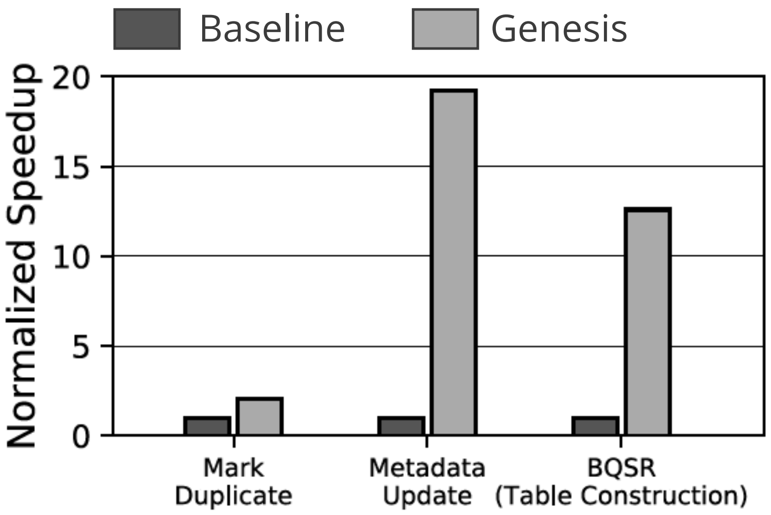
- Genomics and Genome Sequencing
- Genesis: A Hardware Acceleration Framework for Genomic Data Analysis
    - Genesis SQL Frontend
    - Genesis Hardware Library
    - Genesis Backend
    - Genesis-generated HW accelerators
- Evaluation
- Conclusions

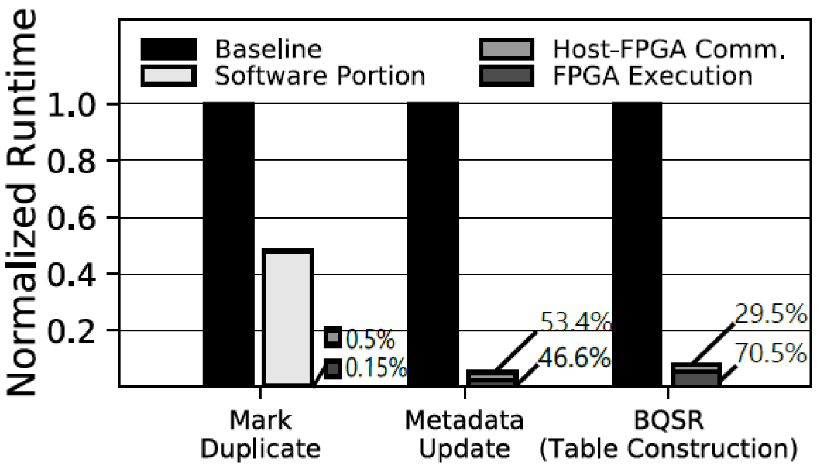
Performance Evaluation

- **Genesis-generated accelerators**: AWS EC2 F1 instance with a single Xilinx VU9P FPGA
- **Baseline**: AWS EC2 instance with 16-threads Intel Xeon Platinum CPU and high-performance SSDs

- Genesis obtains 2 – 18x speedup on three different data manipulation stages
  - Mark Duplicate speedup is bounded by un-accelerated portion
  - Metadata Update and BQSR performance are partly limited by host-FPGA data transfers
    - Speedup will improve with the better interconnect between the host and the FPGA

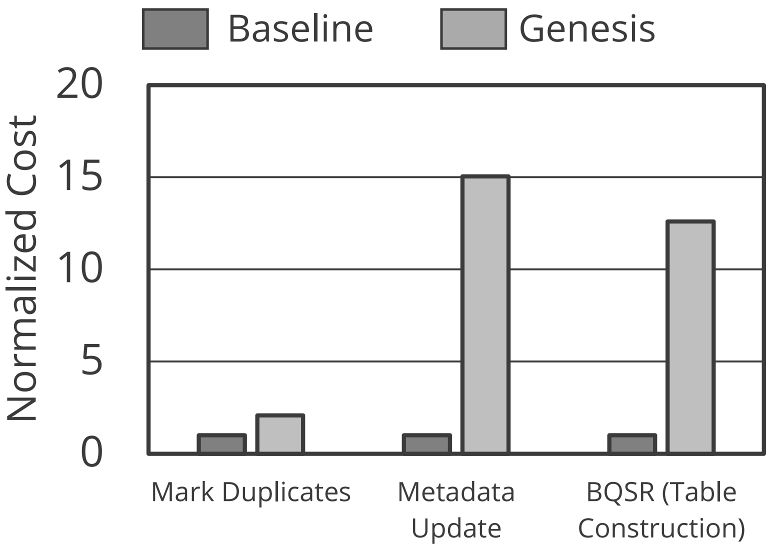
- Speedup can be further improved with the use of multiple FPGAs

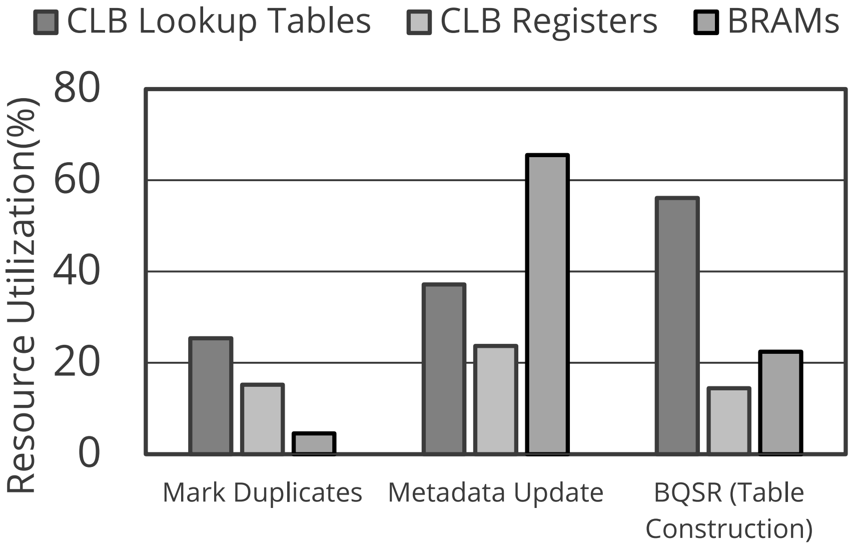




Cost & Resource Usage

- Speedups from Genesis-generated accelerators translate to the cost saving
  - 2-15x cost saving (on AWS EC2 cloud) over the GATK4 baseline
  - Slightly less cost saving than the speedup (80%-99%)
- Different accelerators are bounded by different types of resources
  - Most Used Resource Type
    - *Mark Duplicate & BQSR* – CLB Lookup Tables (Mostly Logic) | *Metadata Update* – BRAMs (SPM)
    - Can co-locate different accelerators on the same FPGA





Conclusion

Genesis frames data-manipulation operations in genome sequencing pipeline as RDBMs operation and aids the designing of hardware accelerators for it in a composable way

- Genesis aids the accelerator design for the data manipulation operations with the followings:
  - **SQL Frontend** users utilize to represent the target data-manipulation operation
  - **Hardware Library** which contains hardware blocks accelerating relational operators as well as a genomics-specific operation
  - **Backend** which automatically augments the pipeline with parallelism, deploys it on cloud FPGA, and allows a user to access it with high-level API
- 2-18x speedup as well as 2-15x cost saving on data manipulation operations in the data preprocessing phase of the GATK4 genome sequencing pipeline

The 47th IEEE International Symposium on Computer Architecture

Genesis: A Hardware Acceleration Framework for Genomic Data Analysis

Tae Jun Ham, David Bruns-Smith, Brendan Sweeney, Yejin Lee,
Seong Hoon Seo, U Gyeong Song, Young H. Oh,
Krste Asanovic, Jae W. Lee, Lisa Wu Wills

SEOUL NATIONAL UNIVERSITY

Berkeley
UNIVERSITY OF CALIFORNIA

SUNG KYUN KWAN UNIVERSITY(SKKU)

Duke
UNIVERSITY