Specializing the Storage System for Genomics:

A Case for Efficient Input Processing in the Storage System

Presenter: Matteo Dietz
Mentor: Nika Mansouri Ghiasi
Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms
Date: May 30, 2024
Duration: 10 Weeks

SAFARI
SAFARI Research Group
safari.ethz.ch

ETH zürich

Executive Summary

Problem: Large I/O-overheads significantly bottlenecking genome analysis pipelines due to low external bandwidth of modern SSDs

Goal: Reduce the I/O-overheads to speed up the pipeline. This allows us to profit from faster accelerators for read mapping without bottlenecking their throughput.

Idea: Perform format conversion inside the SSD to reduce the data movement overhead.

Our Approach: Convert the input file formats from ASCII to 2bit encoding and only send the bases without the quality scores whenever the accelerator requires this.

Key Results: HW: **x3.23** (**x1.36**) speedup for GenCache & **x3.23** speedup for GEM with binary encoded files compared to FastQ files using a SATA (PCIe Gen 4) SSD

SW: **x1.10** (**x1.07**) speedup for Minimap2 with FastA files compared to FastQ files using a SATA (PCIe Gen4) SSD

Talk Outline

Background & Motivation

Processing Inputs in the SSD

Evaluation Results

Conclusion

Next Steps

Talk Outline

Background & Motivation

Processing Inputs in the SSD

Evaluation Results

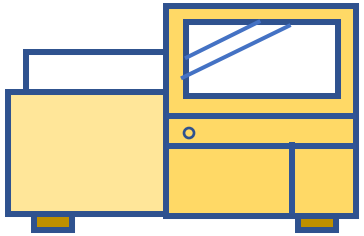
Conclusion

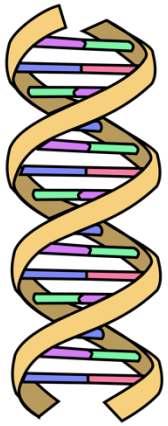
Next Steps

Background

Importance of genome sequence analysis

- Many applications in:
  - Personalized medicine
  - Outbreak tracing
  - Evolutionary studies
- Large amounts of data to be analyzed



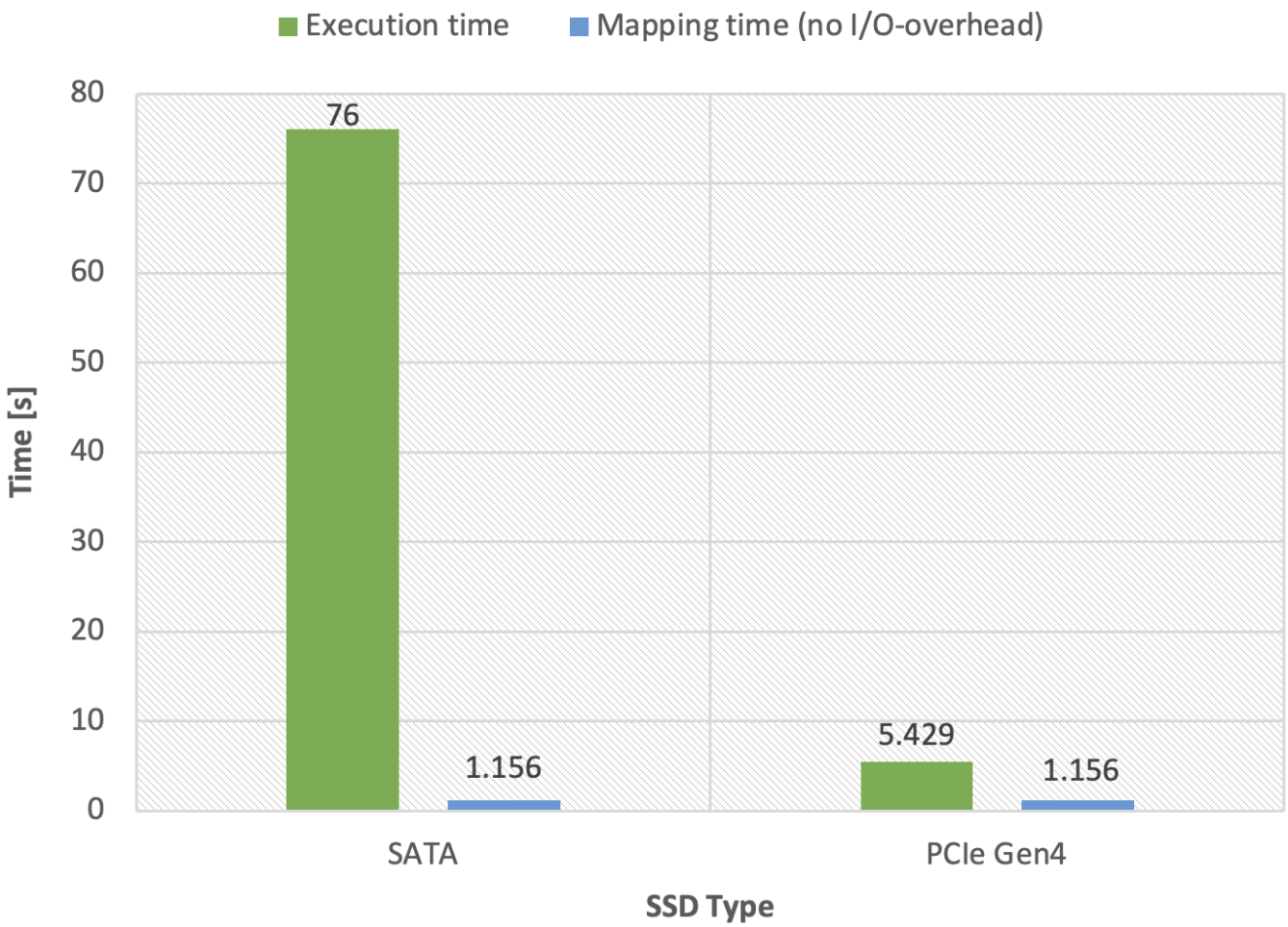


Read mapping

- Comparison of genomic reads to a reference genome
- Computationally expensive (ASM)
- High I/O costs due to unnecessary movement of large amounts of low-reuse data from storage systems to main memory and computation units

Motivation

- **GEM** (2023) for a read set of size 15GB in FastQ format



Goal

Observation: Genome analysis pipeline is significantly bottlenecked by I/O-overheads.

Goal: Reduce I/O-overheads using in-storage file format conversion to increase the throughput of the pipeline.

Talk Outline

Background & Motivation

Processing Inputs in the SSD

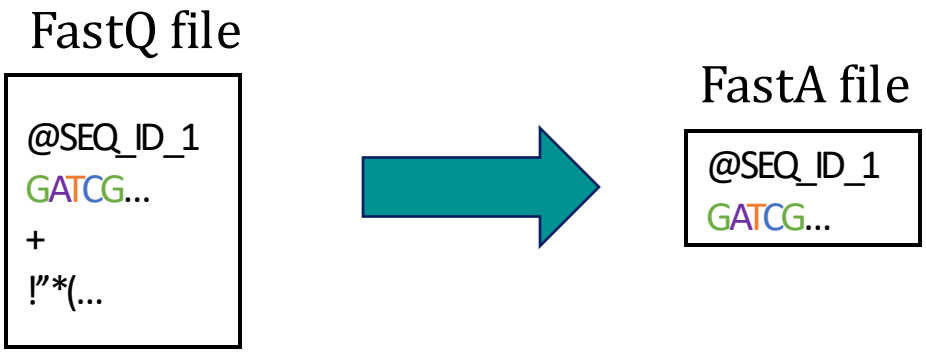
Evaluation Results

Conclusion

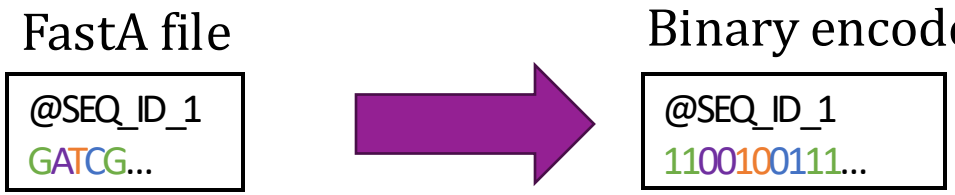
Next Steps

Processing Inputs in the SSD

- **Idea:** Reduce file size of genomic reads inside the SSD to reduce the data movement overhead from the storage system for hardware mappers
  - Send only the bases when the quality score in the read set is not needed

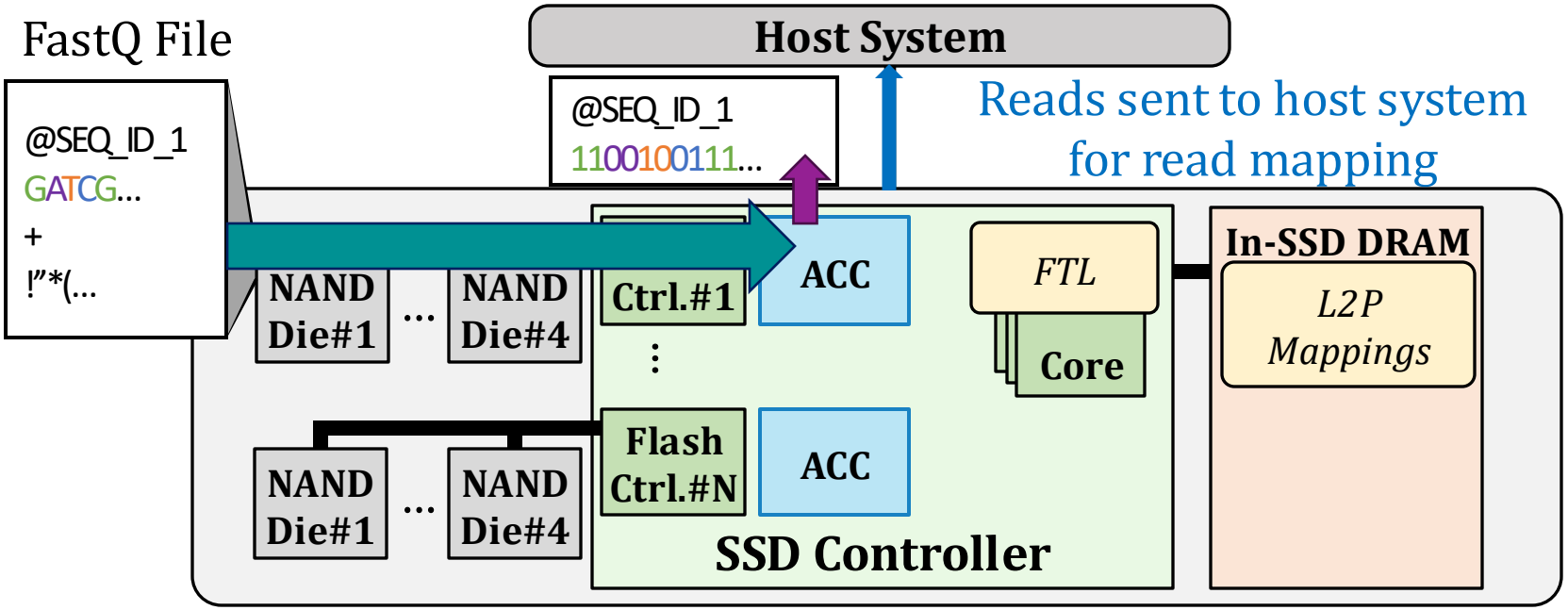


  - Perform format conversion in the SSD to match the format required by the accelerator

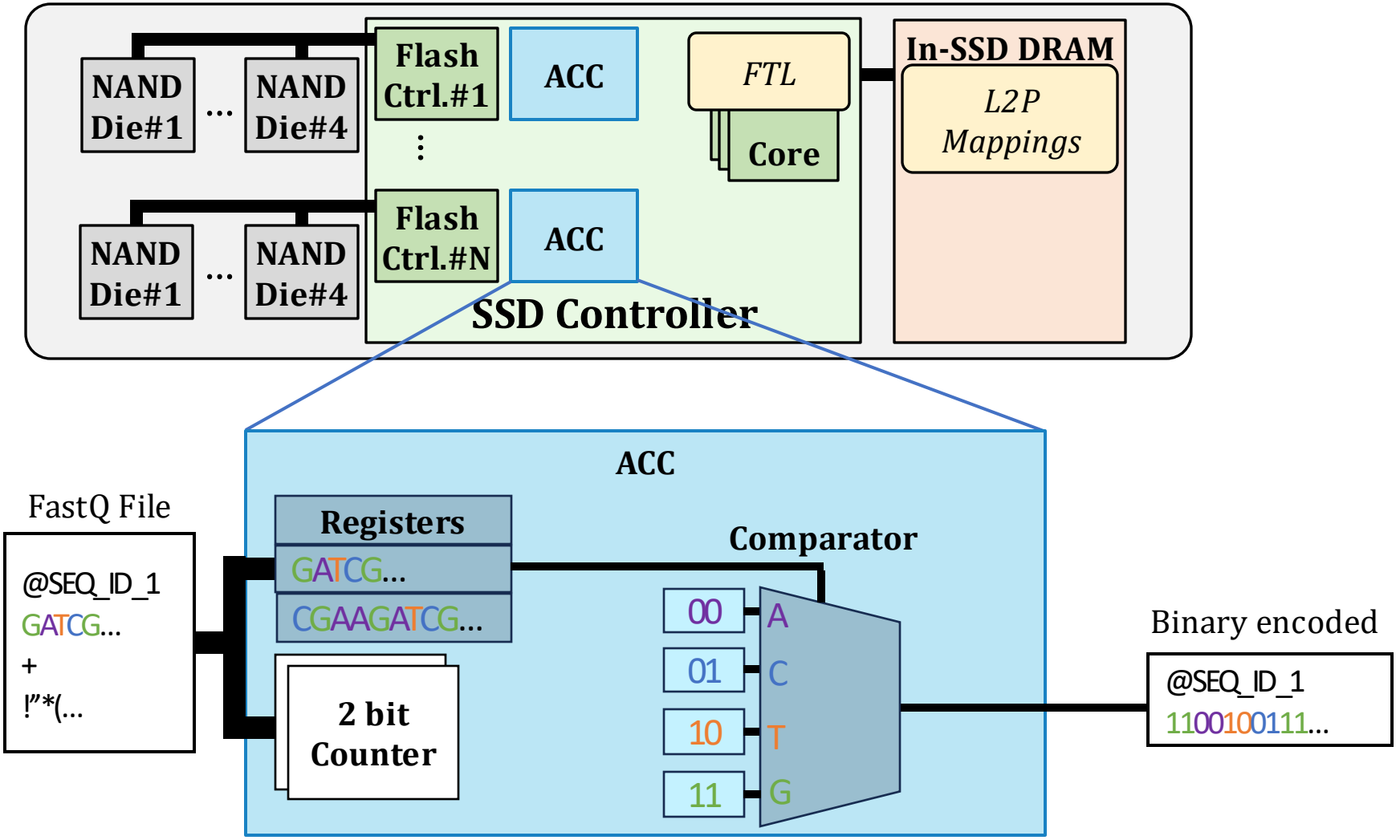


Processing Inputs in the SSD

- **In-storage** input file format conversion to reduce the overhead due to large amount of data movement



Accelerator Design



Talk Outline

Background & Motivation

Processing Inputs in the SSD

Evaluation Results

Conclusion

Next Steps

Evaluation Methodology

Read Mapper

- **HW mapper**: **GenCache** (2019), **GEM** (2023)
- **SW mapper**: **Minimap2** with 128 threads running on bio machine (safari-nexus0)

SSDs

- SATA3 SSD (0.5 GB/s sequential read bandwidth)
- PCIe Gen4 SSD (7 GB/s sequential read bandwidth)

Assumption

- In-storage file format conversion accelerator has higher throughput than external bandwidth of SSD ⟶ does not bottleneck the pipeline

Performance with HW Mapper

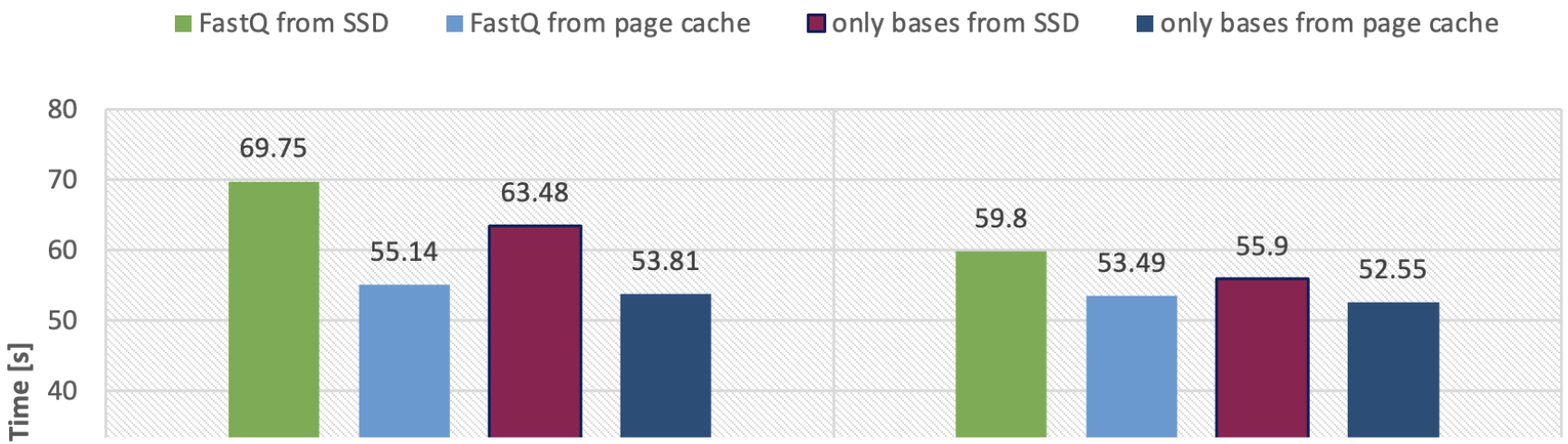
- Left: **GenCache** (2019) for a read set of size 15GB in FastQ format
- Right: **GEM** (2023) for a read set of size 15GB in FastQ format

GenCache achieves a **x3.23** (**x1.36**) speedup with binary encoded input files compared to the full FastQ files when the data is coming from a SATA (PCIe Gen4) SSD

GEM achieves a **x3.23** speedup with binary encoded files compared to the full FastQ files when the data is coming from a SATA or a PCIe Gen4 SSD

Performance with SW Mapper

- **Minimap2** for a read set of size 9.2GB in FastQ format



Minimap2 achieves a **x1.10** (**x1.07**) speedup with FastA files compared to the full FastQ files when the data is coming from a SATA (PCIe Gen4) SSD

Key Takeaways

In-storage file format conversion can significantly increase the throughput of genome analysis pipelines by reducing the unnecesarily large amount of data movement.

Both hardware and software mappers can benefit from these input file conversions. The execution time savings through I/O-overhead reduction are more pronounced for the hardware mappers.

Talk Outline

Background & Motivation

Processing Inputs in the SSD

Evaluation Results

Conclusion

Next Steps

Conclusion

Problem: Large I/O-overheads significantly bottlenecking genome analysis pipelines due to low external bandwidth of modern SSDs

Goal: Reduce the I/O-overheads to speed up the pipeline. This allows us to profit from faster accelerators for read mapping without bottlenecking their throughput.

Idea: Perform format conversion inside the SSD to reduce the data movement overhead.

Our Approach: Convert the input file formats from ASCII to 2bit encoding and only send the bases without the quality scores whenever the accelerator requires this.

Key Results: HW: **x3.23** (**x1.36**) speedup for GenCache & **x3.23** speedup for GEM with binary encoded files compared to FastQ files using a SATA (PCIe Gen 4) SSD

SW: **x1.10** (**x1.07**) speedup for Minimap2 with FastA files compared to FastQ files using a SATA (PCIe Gen4) SSD

Talk Outline

Background & Motivation

Processing Inputs in the SSD

Evaluation Results

Conclusion

Next Steps

Next Steps

For this project:

- Implement the hardware accelerator for file format conversion inside the SSD controller
  - Measure power and area overhead of the accelerator
  - Measure end-to-end energy consumption

More general:

- Identify what other simple operations in genomics can be performed in the SSD to improve the performance of genome sequence analysis

Specializing the Storage System for Genomics:

A Case for Efficient Input Processing in the Storage System

Presenter: Matteo Dietz
Mentor: Nika Mansouri Ghiasi
Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms
Date: May 30, 2024
Duration: 10 Weeks

SAFARI
SAFARI Research Group
safari.ethz.ch

ETH zürich