Zero-Copy Merge with RNTuples

EUGENIO MARINELLI

Supervisors:
Jakob Blomer
Javier Lopez Gomez

Introduction

- Datasets in HEP are often produced in a parallel manner
  - partial results that are merged into a single file at the end of the data processing workflow
- **Merging** the content of many RNTuples requires significant resource consumption
- Modern file systems allow for zero-copy cloning of data blocks for a file
  - potentially **high impact** for the merging of HEP data (no physical copying required)

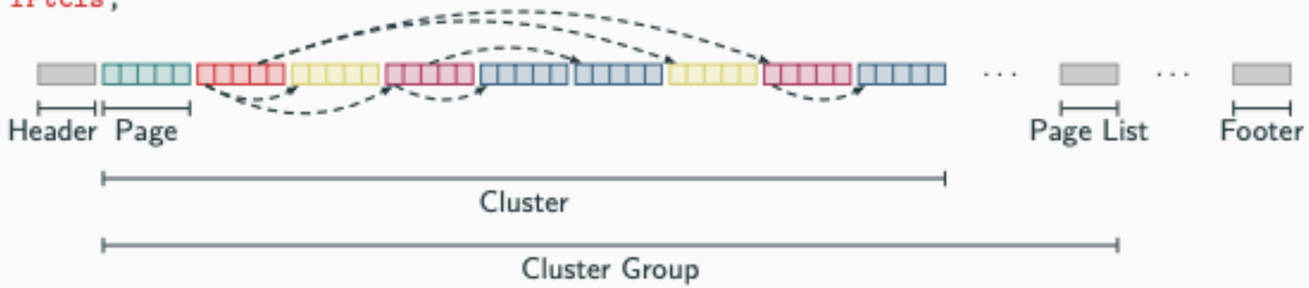
Zero-Copy Merge Project

- In this work:
  - analyze:
    - the zero-copy functionality in modern file systems
    - the requirements to use it with ROOT RNTuple
      - ROOT RNTuple is a new I/O subsystem (still in development)
  - implement a first version of Zero-Copy Merge on ROOT RNTuples
  - comparison with alternative implementations

RNTuple Merging Operation

- RNTuple file:
  - Metadata
  - Events data

```
struct Event {
    int fId;
    vector<Particle> fPtcls;
};
struct Particle {
    float fE;
    vector<int> fIds;
};
```



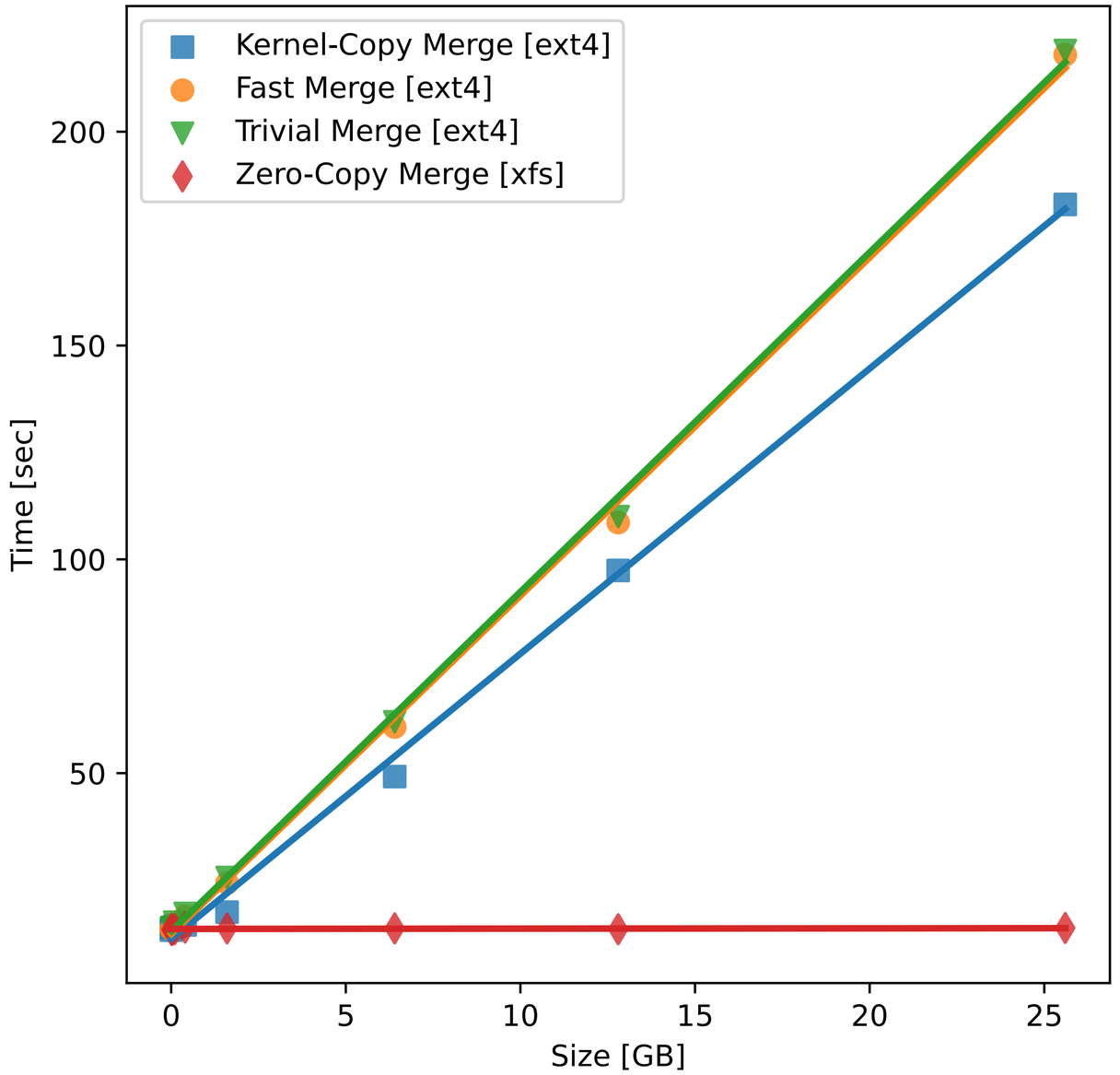
- We want one resulting RNTuple with:
  - Metadata updated
  - Same data content of all the source RNTuples
    - Note: we want to **clone** the content without copying it (zero-copy)

Evaluation

- Zero-Copy merge compared with:
  - Trivial merge
    - read and write data using standard RNTuple API
  - Fast merge
    - read from source files and write on destination file (user space)
  - Kernel merge
    - file copy in kernel space

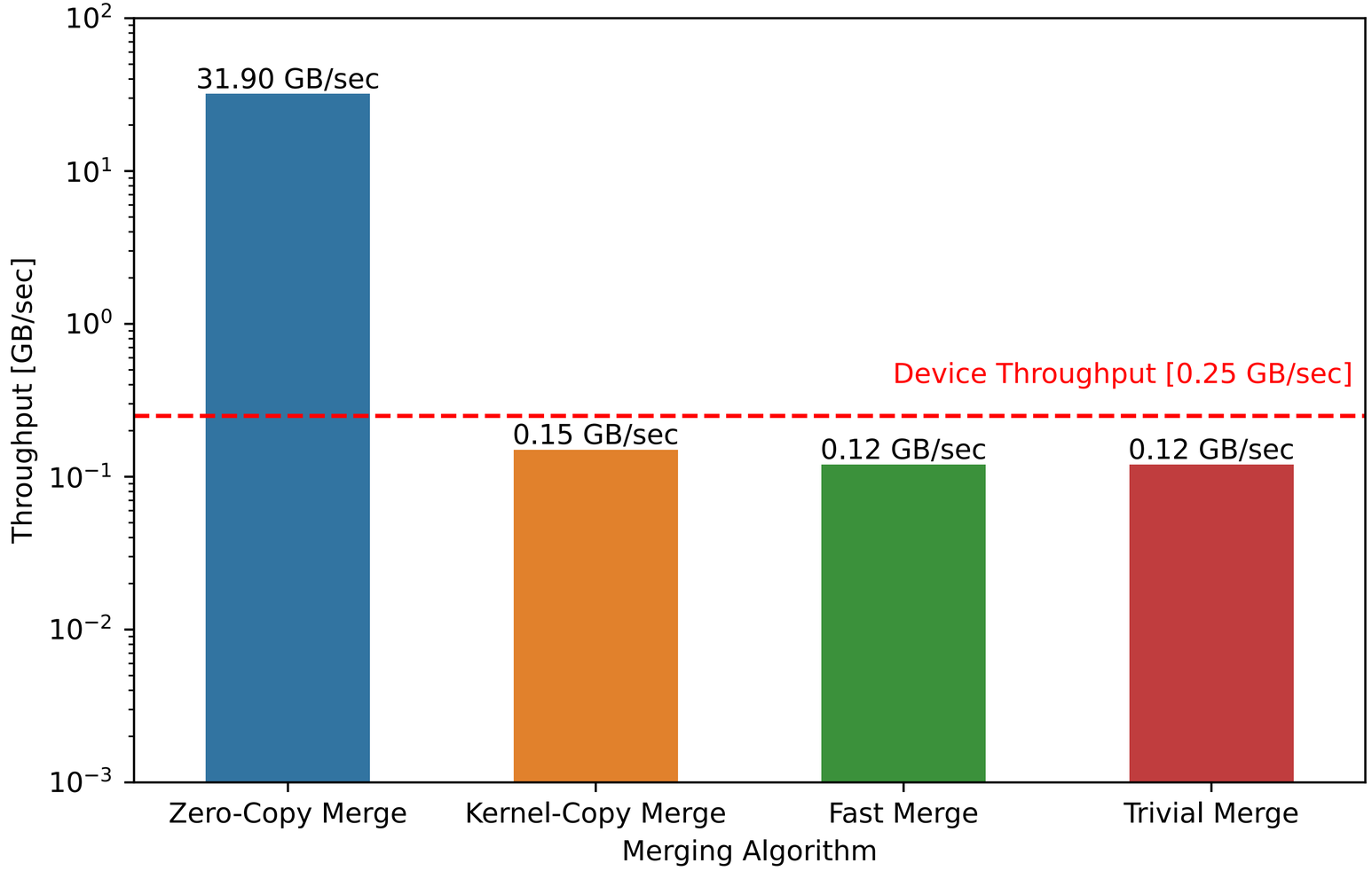
Evaluation: Execution Time

- Zero copy merge nearly constant
  - metadata updates is the only contribution
- Other merge grows linearly
- Test data is uncompressed



Evaluation: Throughput

- We achieve higher throughput than the device limit



Conclusion

- This first Zero-Copy merge implementation enables the RNTuple merge in almost constant time
- These results have significant impact on ROOT's object store support
  - HEP data in object stores (such as Intel DAOS) is by nature prepared to exploit zero-copy merging

THANK YOU!