GranPipe: Composable Hierarchical Pipelines for Near-Data Processing

Johannes Pietrzyk
TU Dresden
Dresden, Germany
johannes.pietrzyk@tu-dresden.de

Dirk Habich
dirk.habich@tu-dresden.de
TU Dresden
Dresden, Germany

Wolfgang Lehner
wolfgang.lehner@tu-dresden.de
TU Dresden
Dresden, Germany

Philippe Bonnet
bonnet@di.ku.dk
Københavns Universitet
Copenhagen, Denmark

Modern data-intensive applications are confronted with rapidly increasing volumes of data, posing significant challenges for efficient and timely processing. The traditional compute-centric approach suffers significantly from the *storage wall*, a critical performance bottleneck caused by the disparity of available interconnect bandwidth and processing power. Near-data processing (NDP) describes a promising approach addressing this storage wall by moving computation closer to data storage locations to minimize data movement across system interconnects. Recent developments in NDP, such as SmartSSD or computational storage devices, have demonstrated substantial performance improvements by offloading compute-intensive tasks, including filtering, summarization, directly onto storage or memory hardware. Despite these promising results, fully exploiting NDP requires sophisticated programming frameworks that explicitly support granular data partitioning and flexible operation offloading, something current frameworks partially address but do not comprehensively solve.

To bridge this gap, we introduce **GranPipe** and share our initial ideas behind it. **GranPipe** is a structured, composable pipeline framework specifically designed to separate data access and traversal from processing. At its core, **GranPipe** provides non-owning hierarchically structured data views for given granularities such as blocks, chunks, pages, and individual data elements. Users can inject specialized processing schemes at every level of the hierarchy to transform the current part of the data. Each transformation stage can be injected independently and composed declaratively at compile time, allowing for reuse, recombination, and reconfiguration without entangling implementation details (see code listing). This composability extends to merging multiple **GranPipe** instances into coordinated, lockstep pipelines, facilitating binary operations across disjoint datasets or execution contexts (lines 5-8 in code). Such flexibility supports a wide range of use cases, from in-memory analytics to near-data offloading, while preserving maintainability, type safety, and performance through zero-cost abstractions.

```
1 auto pipeline =
2   GranPipe::create<exec::host>(std::cout, element_fn{})
3     .at_block_level<exec::device>(parse_fn{})
4     .at_chunk_level<exec::device>(filter_fn{})
5     .merges_with(
6       GranPipe::create<exec::host>(...)
7         ...
8     );
9 pipeline("lineitem.parquet");
```

Moreover, **GranPipe** naturally aligns with the NDP paradigm by explicitly defining computational tasks suitable for offloading to near-data processing devices. **GranPipe** transparently translates logical data partitions through lightweight metadata annotations into actionable offloading directives, enabling asynchronous or synchronous execution on remote NDP-capable hardware. Operations such as (a) chunk-level filtering or (b) page-level indexing can thus be executed on near storage devices, greatly reducing data transfer requirements and significantly enhancing overall performance. In summary,

- **GranPipe** supports synchronized and parallel execution through lockstep binary co-processing functors, enabling efficient parallel processing and maximizing data locality.
- **GranPipe** offers a unified programming abstraction for modular, granularity-aware data processing pipelines capable of leveraging NDP architectures. The overall design makes **GranPipe** particularly suited to exploit heterogeneous hardware environments.

Based on these characteristics, we believe that **GranPipe** closes a critical gap between logical data partitioning and physical offloading, offering a practical foundation for building high-performance, heterogeneous analytics systems that minimize data movement and maximize locality-aware execution.

ACKNOWLEDGMENTS. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

VLDB Workshop Reference Format:
Johannes Pietrzyk, Dirk Habich, Wolfgang Lehner, and Philippe Bonnet. GranPipe: Composable Hierarchical Pipelines for Near-Data Processing. VLDB 2025 Workshop: Third International Workshop on Composable Data Management Systems.

Proceedings of the VLDB Endowment. ISSN 2150-8097.