

DAQ/FLES Data Replay

VT25 and in general

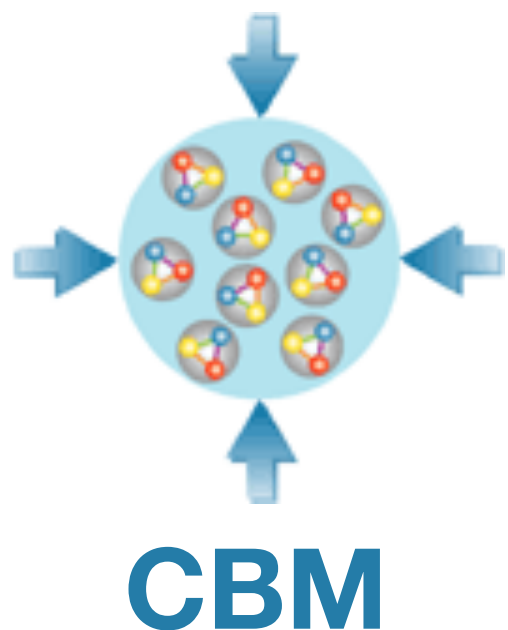
Jan de Cuveland

cuveland@compeng.uni-frankfurt.de

Dirk Hutter

hutter@compeng.uni-frankfurt.de

FIAS Frankfurt Institute for Advanced Studies
Goethe-Universität Frankfurt am Main, Germany

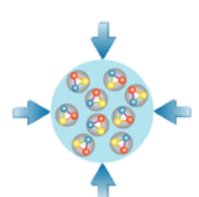


CBM Online Meeting
2025-12-18

Timeslice replay for data challenges

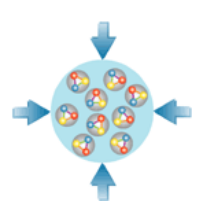
- 3 setups of increasing complexity
 - Use the least complex setup in which you can measure the desired parameters

	Local file source	Shared memory source (fed from local file)	Timeslice forwarding from FLES-IN + shared memory source
Technical setup	Start <code>cbmreco</code> with parameter <code>-i file://...</code>	Start <code>tsclient</code> to read from file(s) and write to shared memory. Start <code>cbmreco</code> with parameter <code>-i shm://...</code>	Distributed setup involving several nodes and applications
Preparation	Copy .tsa files to local volumes on Virgo nodes	Copy .tsa files to local volumes on Virgo nodes	Copy .tsa files to FLES-IN nodes, prepare transmission scheme
Default progress mode	data-based (fully utilize resources)	time-based (limited to fixed timeslice rate)	time-based (limited to fixed timeslice rate)
Can log to InfluxDB?	Yes	Yes	Yes
Can measure timing without disk read / deserialization overhead?	Yes	Yes	Yes



Timeslice forwarding

- There are 2+ options for timeslice forwarding
- Used previously: online monitoring interface based on ZeroMQ publish/subscribe
 - Various known limitations (configuration, monitoring, flow control)
 - Proven to work, but not to be used in final setup → nothing new to learn
- New timeslice forwarding software by ZIB
 - Recently developed, not yet installed at (m)FLES/(m)CBM



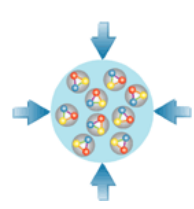
Recent developments at ZIB (F. Schintke)

- **Work on timeslice forwarding ongoing**

- Proof-of-Concept implementation of a central manager and dynamic nodes
 - Using Libfabric, IB and TCP providers
- Interface via shared memory to existing components implemented (both ends)
- Successful forwarding of timeslices with flexible buffer management
- General code improvements, including better error reporting for users

- **Next goals and timelines**

- Preparations for more realistic benchmarks using the size distributions of actual mCBM runs (based on timeslice/microslice metadata)
- Throughput measurements of timeslice forwarder (first results by November)
- Scaling tests and elaboration of the developed flexible buffer management (by November)
- Reconstruction of the complete data flow chain at ZIB, starting from microslice streams (by end of 2025):
 - `Replay/PatternGen --SHMEM--> EntryNodes --Flesnet--> BuildNodes --TS-forwarder--> (SHMEM) ProcessingNodes`



FLES-IN Status

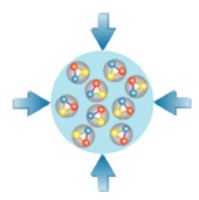
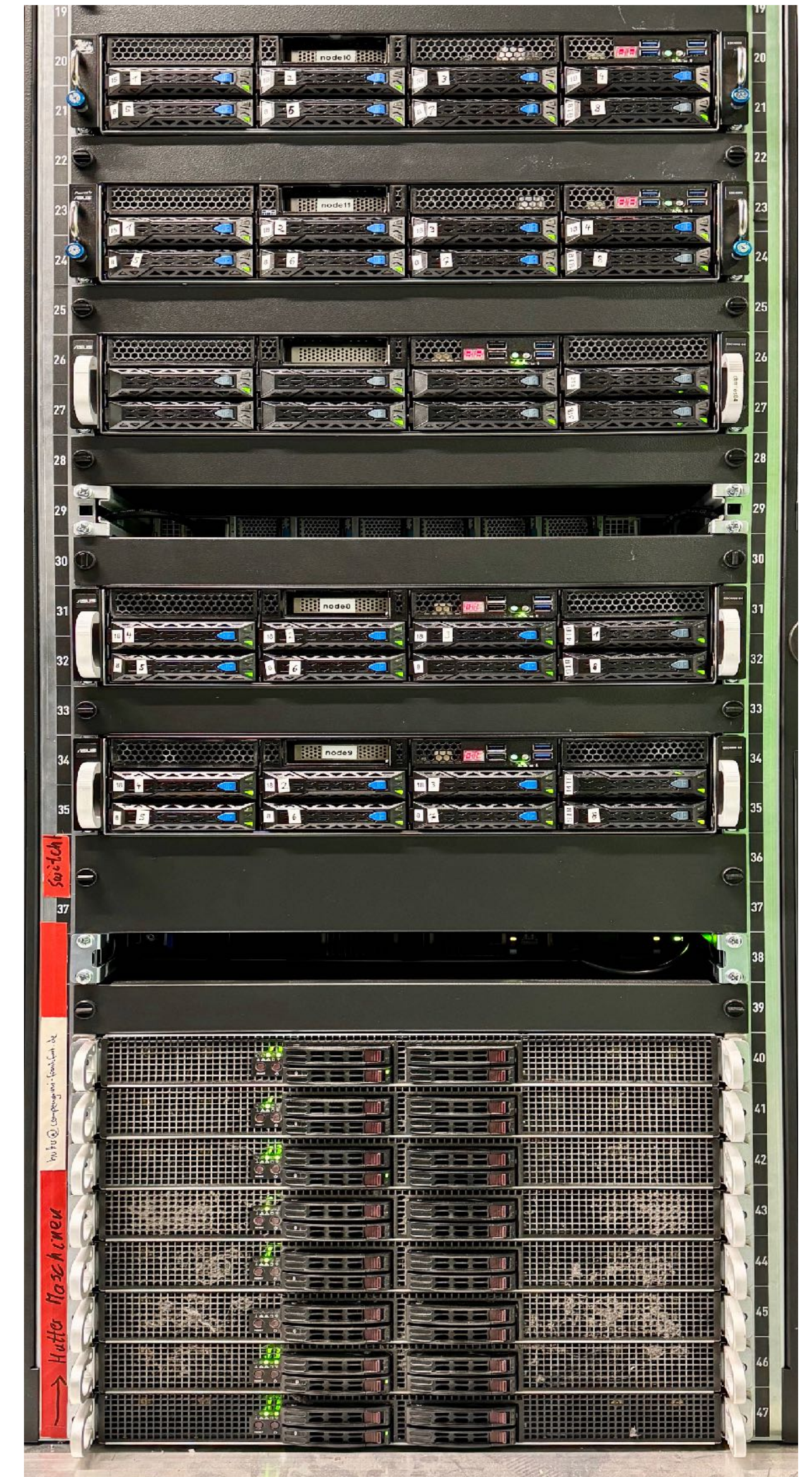
- 4 modern nodes (12 total)
 - up to 32 cores/64 threads, 256 GB RAM
 - Head node for infrastructure services and login
 - 200 GB/s InfiniBand HDR to mCBM and Virgo
- FLES-IN storage space
 - 4 nodes (node[8-11]) with each:
 - 32 TB SSD (2x 16 TB)
 - 60 TB HDD (2x 18 TB + 3x 8 TB)
 - **368 TB total**
 - 128 TB SSD
 - 240 TB HDD
- Current (2025-12-17) storage state
 - SSDs: 57% used
 - HDDs: ~80% used
 - Mostly mCBM data of runs between 2024-06 and 2025-05
 - What can we delete?

2nd gen
Processing Nodes
with local storage

Head Node

2nd gen
Processing Nodes
with local storage

1st gen
Processing Nodes



mFLES components and connectivity

