ExSeisPIOL: Extreme-Scale Parallel I/O for Seismic Workflows

Parallel I/O Challenges for Seismic

- Mixture of I/O patterns. e.g
  - Consecutive access of traces
  - Non-monotonic access, non-contiguous, non-consecutive access
- MPI usage common in O&G codes: MPI and MPI-IO limits → ~2 GiB per call
- Collective I/O → balancing of calls on each process.
  - Mismatch → deadlock

Parallel I/O Challenges for Seismic (contd)

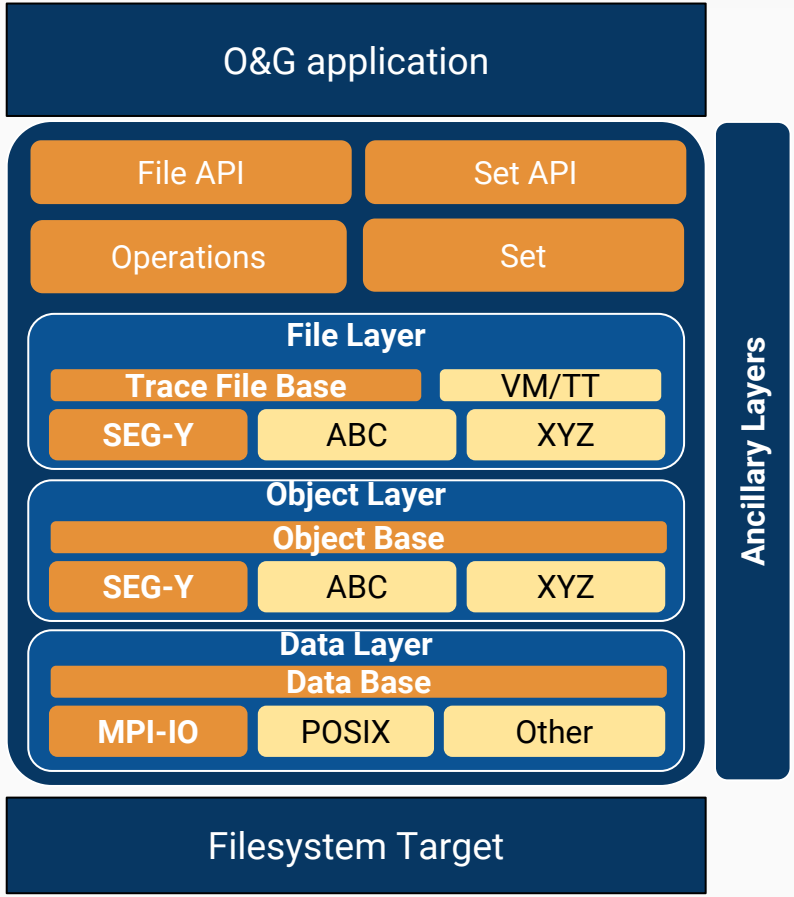
- Conformance to the SEG-Y standard is variable.
- Trace data may be stored in obsolete IBM floating point format.
- Difficult to have all of readability, maintainability and scaling/performance of seismic processing codes without substantial effort!

ExSeisPIOL

- **Ex**treme-Scale **Seis**mic **P**arallel **I/O** **L**ibrary (**ExSeisPIOL**)
- Easy to use, geophysicist-friendly C++ and C APIs
- Scalable / Performance
- Reduces maintenance → Reduces codebase sizes substantially
- Extensive testing framework (unit, integration and system tests)

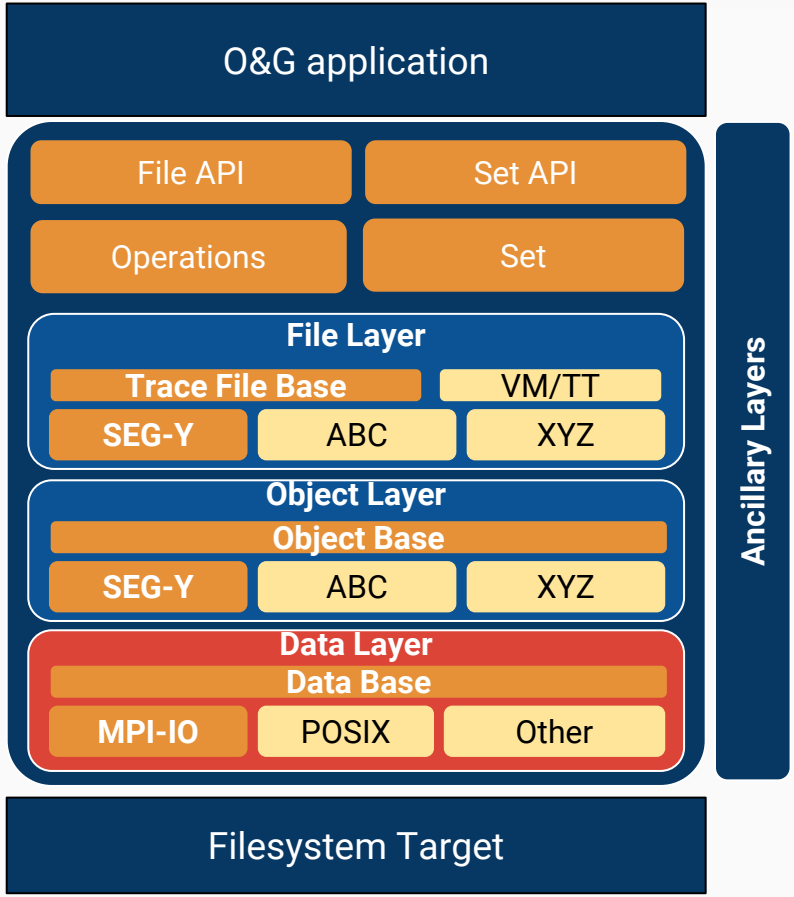
ExSeisPIOL Design

- Multi-Layer solution → separate file-format processing, layout and MPI-IO details.
- Two public APIs:
  - High-level set API: Implicit I/O
  - Low-level file API: Explicit I/O
- Computational geophysicists / software engineers writing seismic processing software on HPC clusters



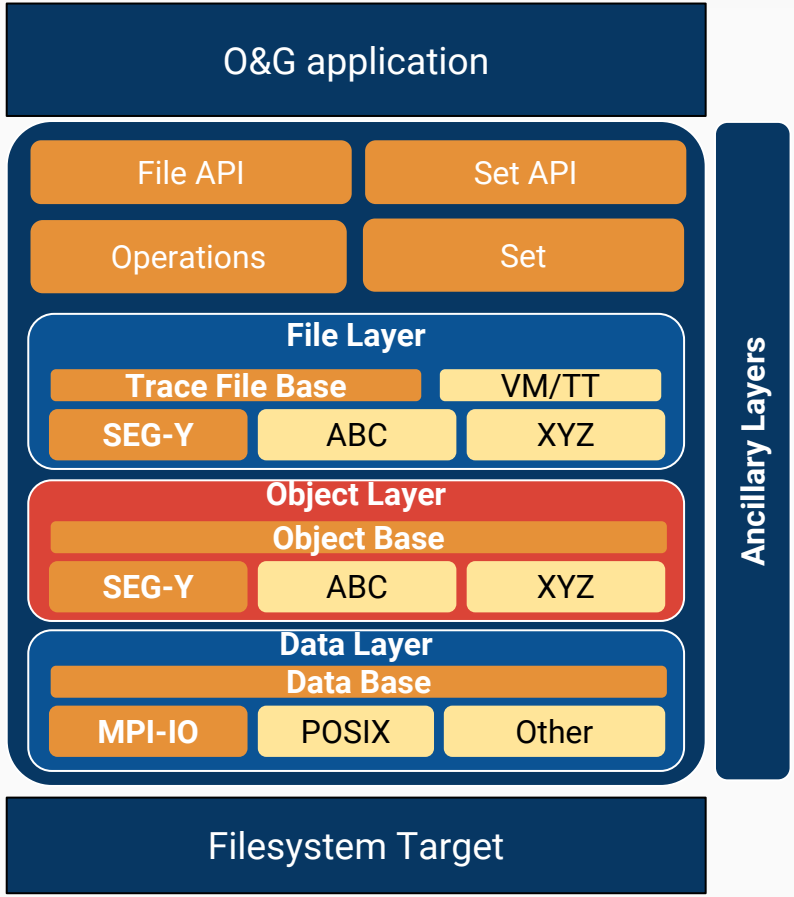
ExSeisPIOL Design

- Multi-Layer solution → separate file-format processing, layout and MPI-IO details.
- Two public APIs:
  - High-level set API: Implicit I/O
  - Low-level file API: Explicit I/O
- Computational geophysicists / software engineers writing seismic processing software on HPC clusters



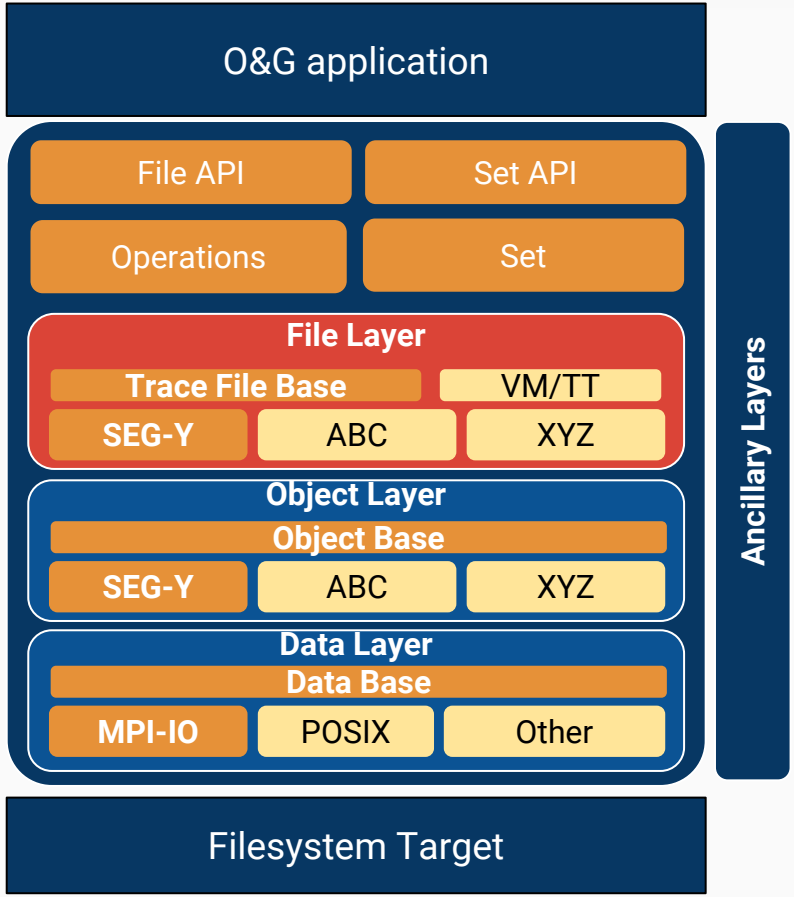
ExSeisPIOL Design

- Multi-Layer solution → separate file-format processing, layout and MPI-IO details.
- Two public APIs:
  - High-level set API: Implicit I/O
  - Low-level file API: Explicit I/O
- Computational geophysicists / software engineers writing seismic processing software on HPC clusters



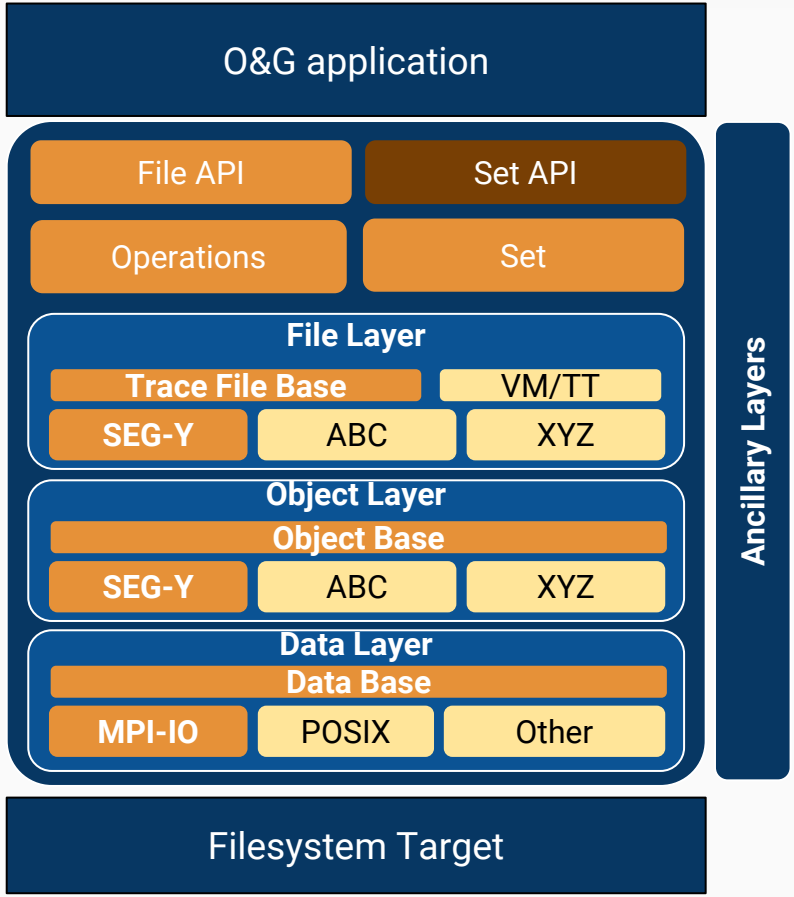
ExSeisPIOL Design

- Multi-Layer solution → separate file-format processing, layout and MPI-IO details.
- Two public APIs:
  - High-level set API: Implicit I/O
  - Low-level file API: Explicit I/O
- Computational geophysicists / software engineers writing seismic processing software on HPC clusters



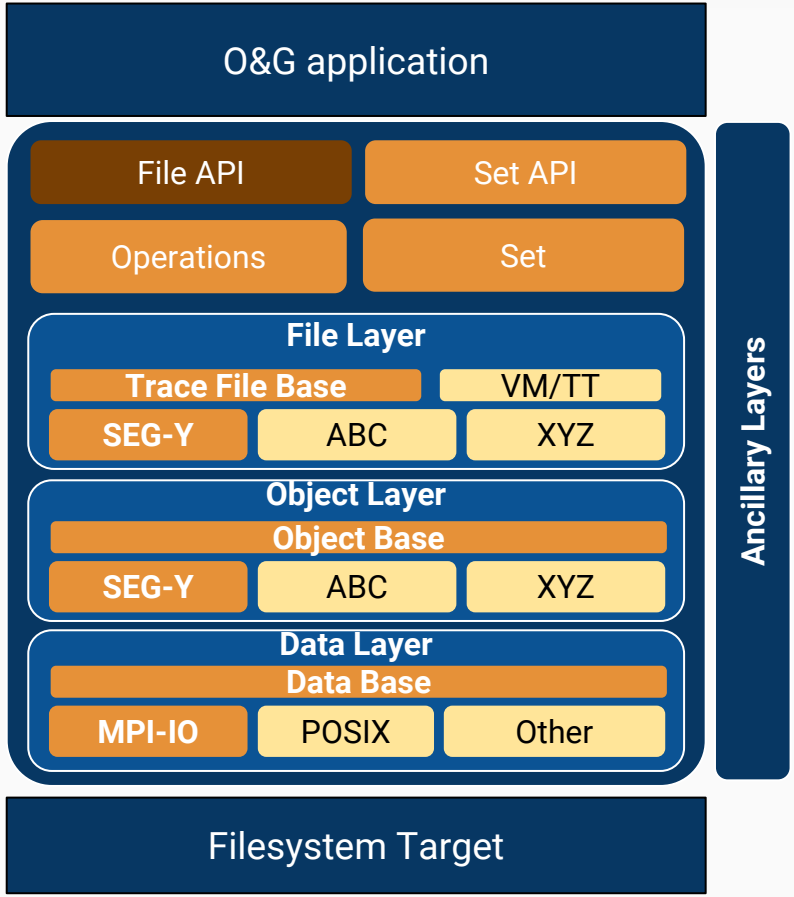
ExSeisPIOL Design

- Multi-Layer solution → separate file-format processing, layout and MPI-IO details.
- Two public APIs:
  - High-level set API: Implicit I/O
  - Low-level file API: Explicit I/O
- Computational geophysicists / software engineers writing seismic processing software on HPC clusters



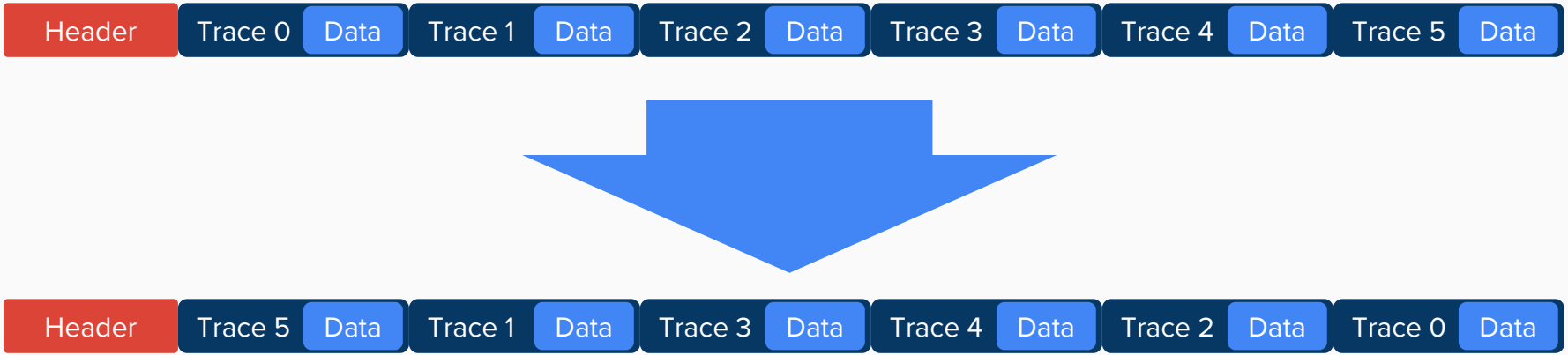
ExSeisPIOL Design

- Multi-Layer solution → separate file-format processing, layout and MPI-IO details.
- Two public APIs:
  - High-level set API: Implicit I/O
  - Low-level file API: Explicit I/O
- Computational geophysicists / software engineers writing seismic processing software on HPC clusters



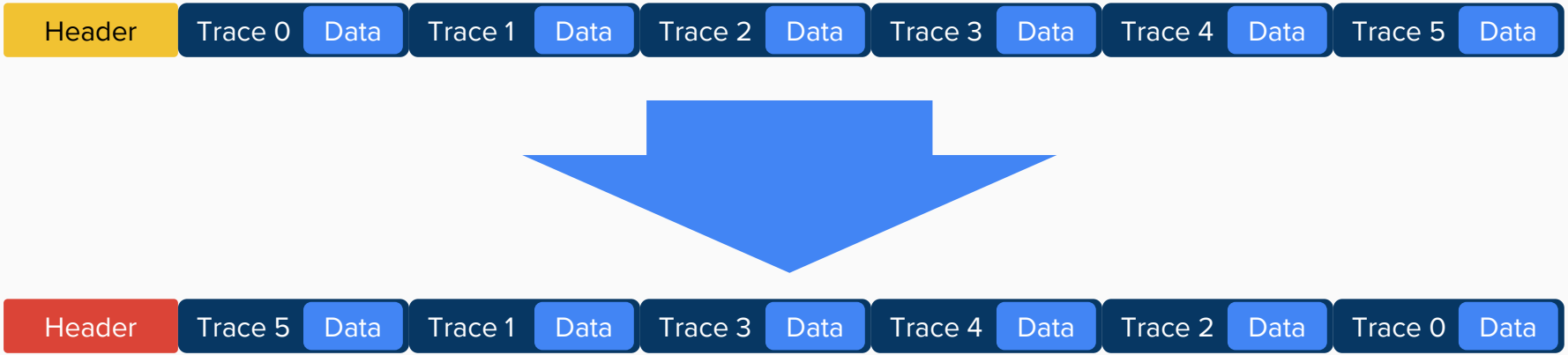
Sorting a SEGY File using Burst-Buffers

A segy file contains a header and a sequence of data which describes the trace coordinates etc and the trace data itself:



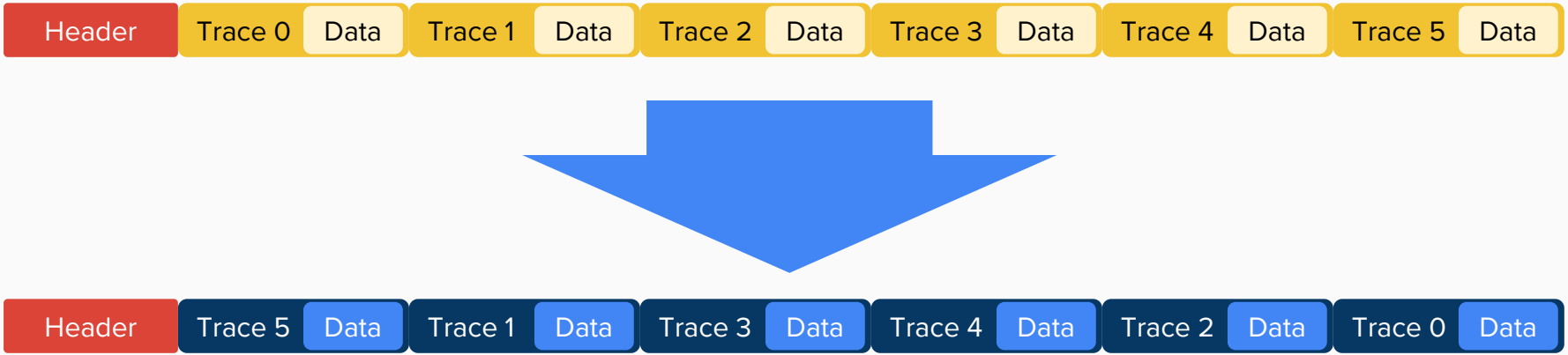
Sorting a SEGY File using Burst-Buffers

A segy file contains a header and a sequence of data which describes the trace coordinates etc and the trace data itself:



Sorting a SEGY File using Burst-Buffers

A segy file contains a header and a sequence of data which describes the trace coordinates etc and the trace data itself:



Sort: Complete Example (C++)

```
#include "set.hh"
using namespace PIOL;
int main(void)
{
    ExSeis piol;
    Set set(piol, "input*.segy", "output");
    set.sort(SortType::OffLine);
    return 0;
}
```

Sort: Complete Example (C++)

```
#include "set.hh"
using namespace PIOL;
int main(void)
{
    ExSeis piol;
    Set set(piol, "input*.segy", "output");
    set.sort(SortType::OffLine);
    return 0;
}
```

Initialise MPI and any ancillary object used by the PIOL.

Sort: Complete Example (C++)

```
#include "set.hh"
using namespace PIOL;
int main(void)
{
    ExSeis piol;
    Set set(piol, "input*.segy", "output");
    set.sort(SortType::OffLine);
    return 0;
}
```

Create a set of traces from every file that matches the “input*.segy” wildcard.

Sort: Complete Example (C++)

```
#include "set.hh"
using namespace PIOL;
int main(void)
{
    ExSeis piol;
    Set set(piol, "input*.segy", "output");
    set.sort(SortType::OffLine);
    return 0;
}
```

Sort the file into Offset, Inline, Crossline order. Only trace headers are read.

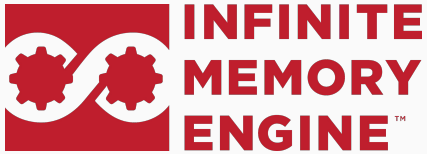
Sort: Complete Example (C++)

```
#include "set.hh"
using namespace PIOL;
int main(void)
{
    ExSeis piol;
    Set set(piol, "input*.segy", "output");
    set.sort(SortType::OffLine);
    return 0;
}
```

(implicit destructor): All substantial I/O for the sort is now performed to create the output file. Implicit MPI deinitialize.

DDN Storage IME: Burst-Buffer Technology

- ExSeisPIOL provides hardware specific optimisations
  - So geophysicists don’t have to!
- DDN IME is a next-generation tiered data-storage architecture
- High-throughput SSDs
- Smart IME Software:
  - Non-contiguous write performance ≈ contiguous write performance



Burst Buffer: I/O Workflow

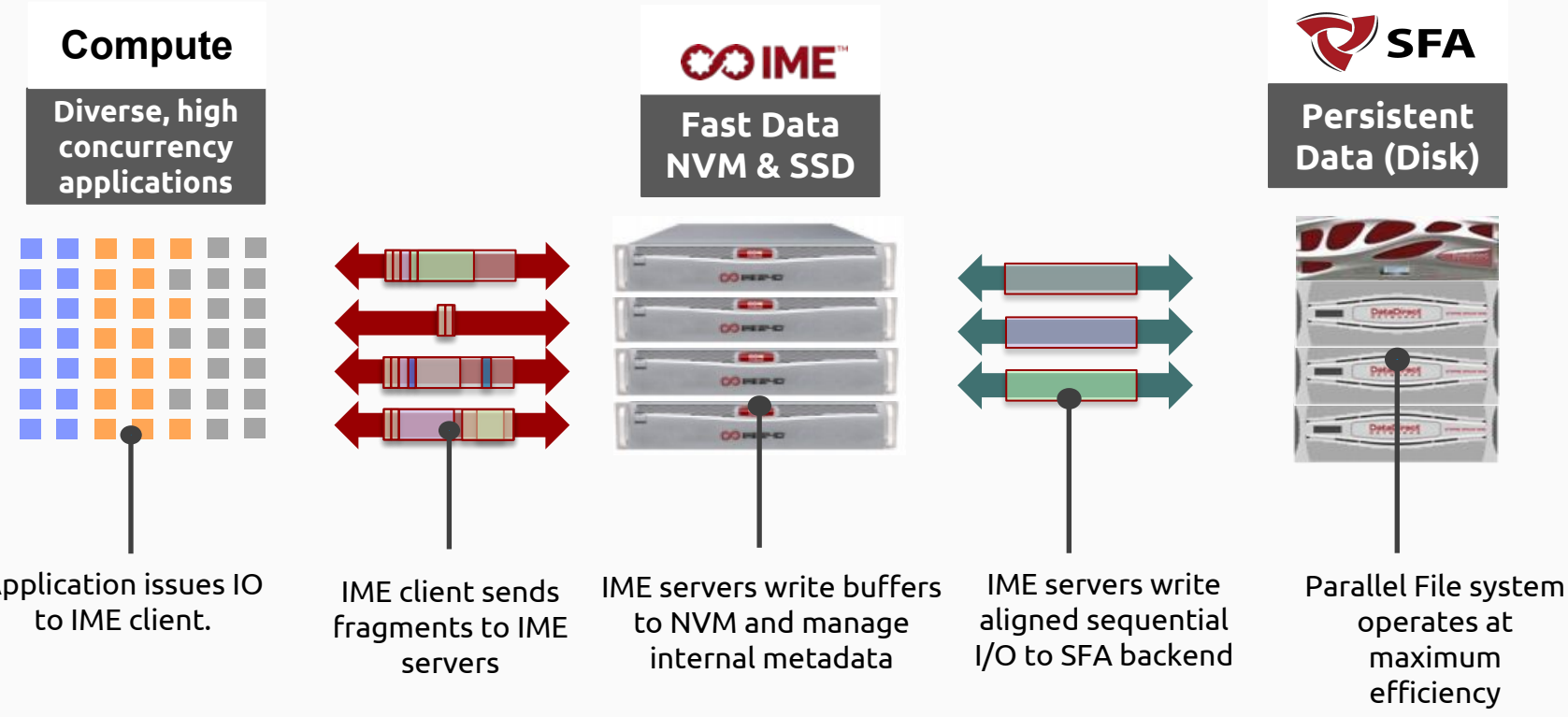
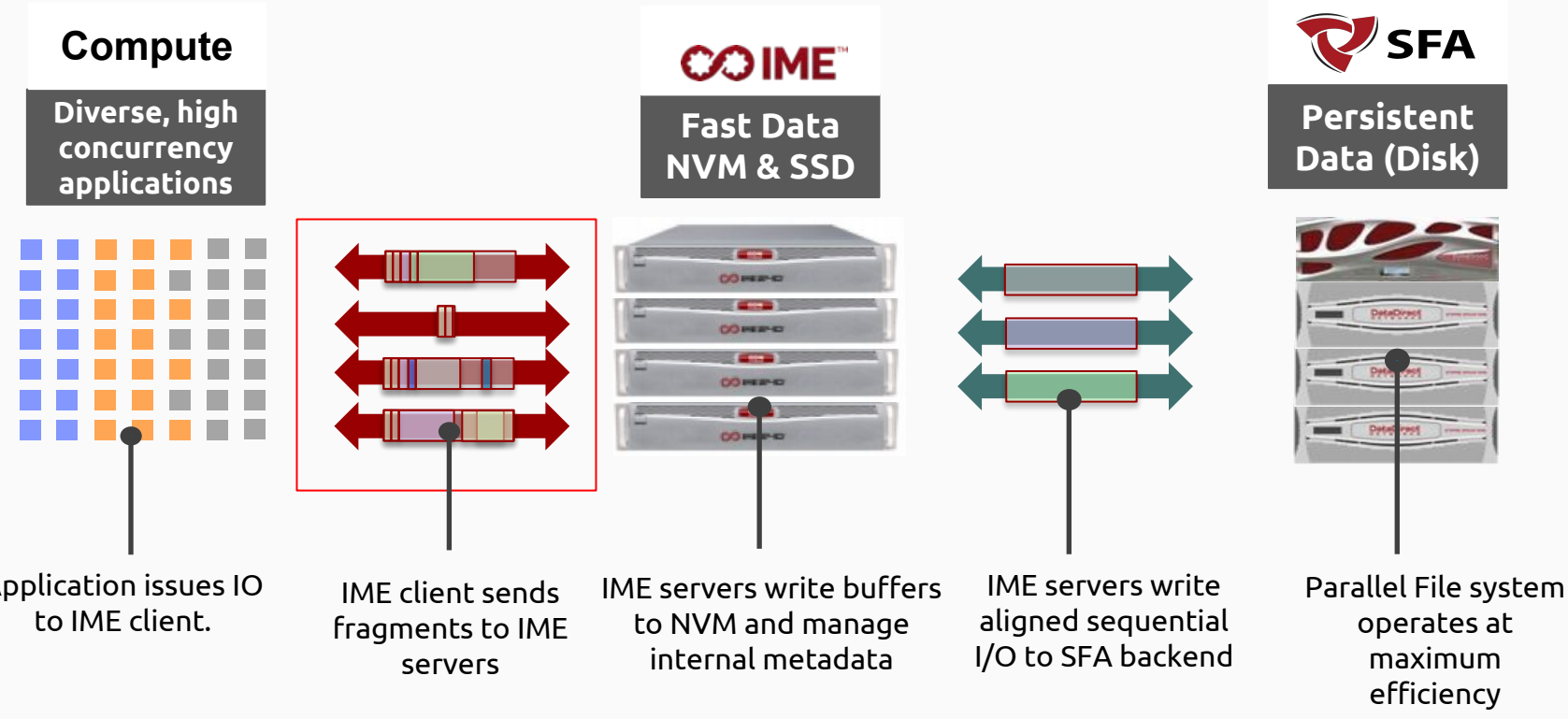
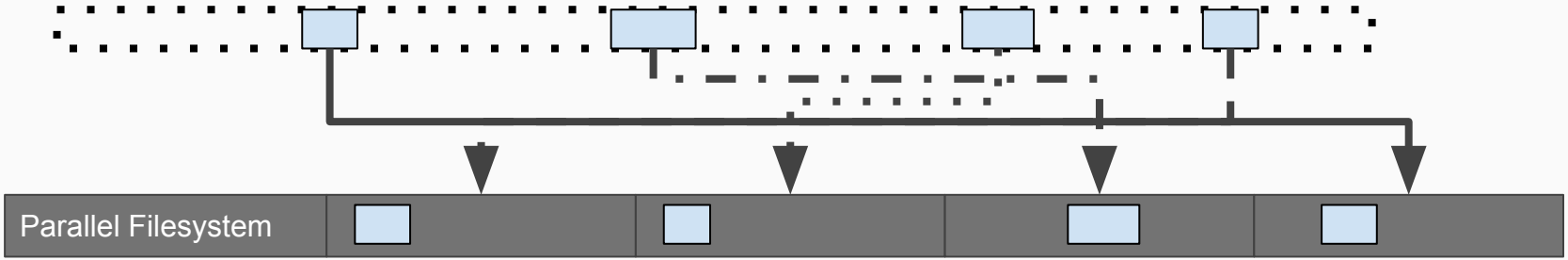


Image courtesy of DDN Storage.

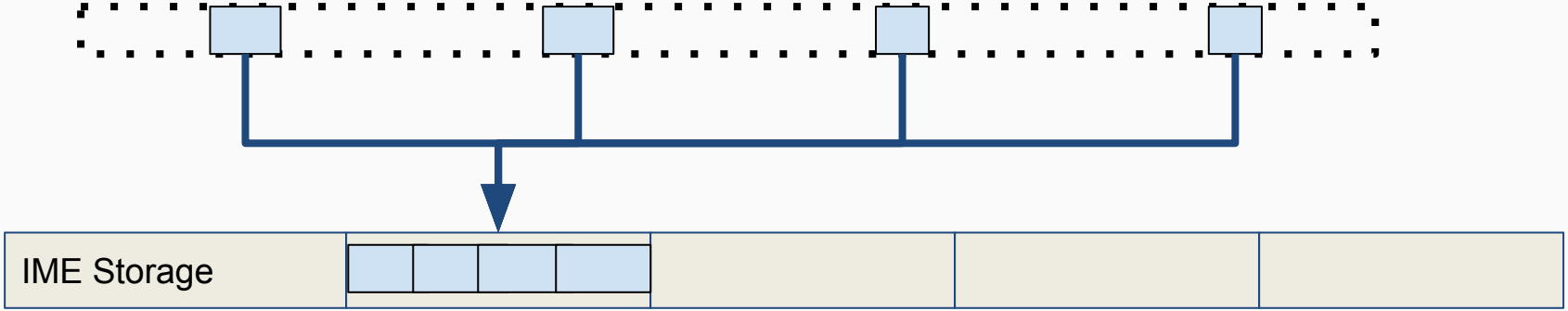
Burst Buffer: I/O Workflow



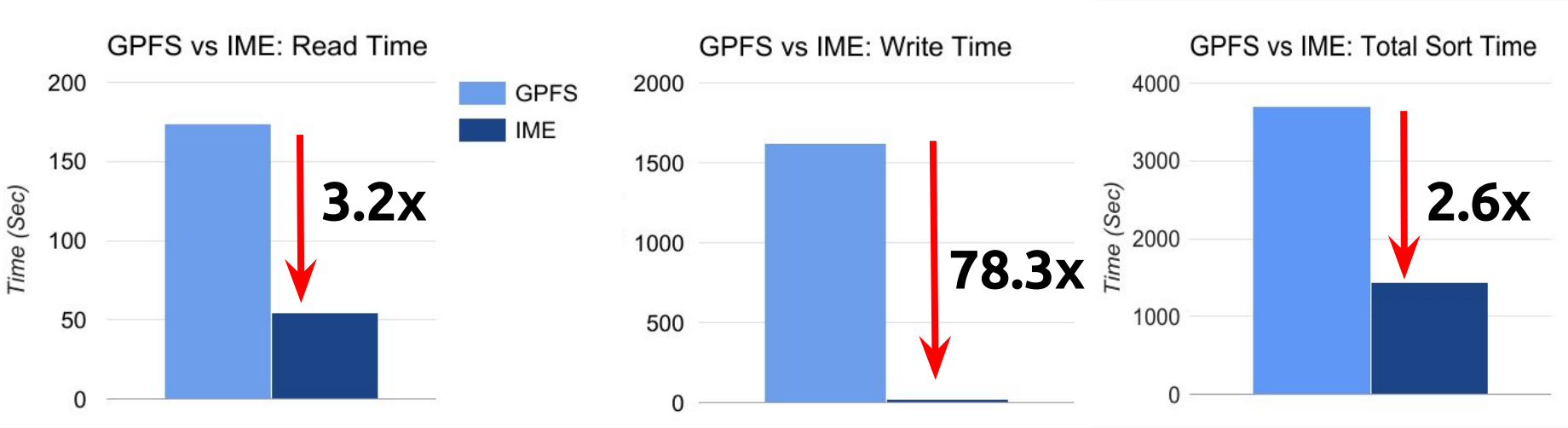
Treatment of I/O patterns, Parallel File-system vs Burst Buffer



IME: Contention free and less redundant reads



SEG-Y Sort Benchmarking - DDN Test Platform



- Comparing GPFS against an IME setup.
- Sort of a 400 GiB SEG-Y file (400 GiB read, 400 GiB write), 4 nodes.
- Total Time: 63% GPFS I/O, only 5% IME I/O for sort
- DDN IME is more than a SSD bank! **Software advantage** over GPFS

Conclusions

- Parallel I/O Library for seismology workflows
- SEG-Y compatible
- ExSeisPIOL can use DDN IME hardware for big speedups
- Easy to use API → increases productivity
- Production ready but development ongoing
- Open source → September 2017

Thanks for listening

Cathal Ó Broin [a,b], Ruairi Short [a], Meghan Fisher [a,b], Seán Delaney [c], Steven Dagg [c], Gareth O'Brien [c], Jean-Thomas Acquaviva [d], Michael Lysaght [a,b]

a. ICHEC, Dublin Ireland
b. Lero, Dublin, Ireland
c. Tullow Oil, Dublin, Ireland
d. DDN Storage, Paris, France

Funding: