Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# OTFX
# An In-memory Event Tracing Extension to the Open Trace Format 2

Michael Wagner[1,2], Andreas Knüpfer[2], Wolfgang E. Nagel[2]

**michael.wagner@bsc.es**

1) Barcelona Supercomputing Center (BSC), Barcelona, Spain
2) Center for Information Services and High Performance Computing (ZIH), Dresden, Germany

# Outline

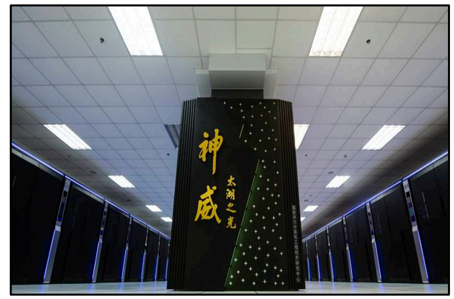
- Introduction
- Concepts for In-memory Event Tracing
- Hierarchical Memory Buffer
- Evaluation
- Conclusion

# High Performance Computing

| 2006 | 2011 | 2016 | 2021 |
|---|---|---|---|

BlueGene/L

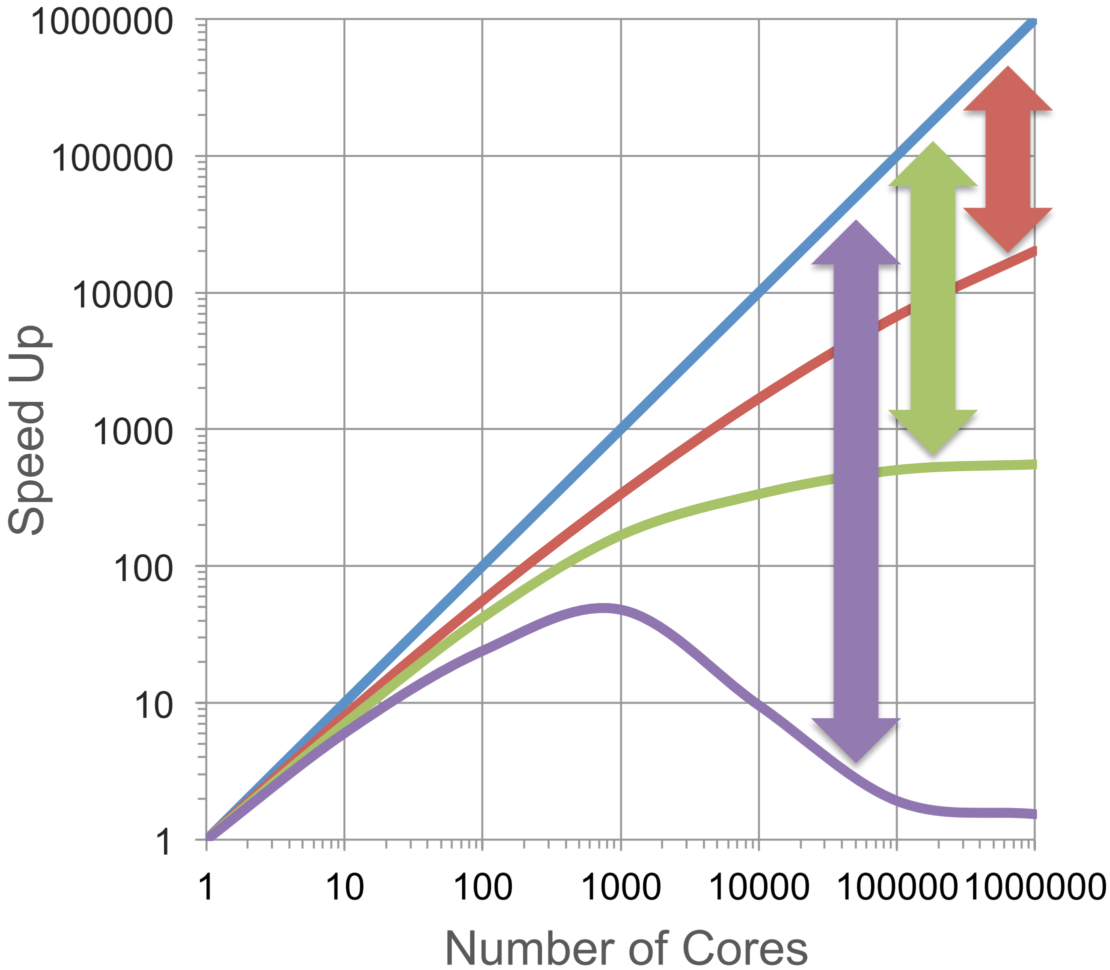

K Computer



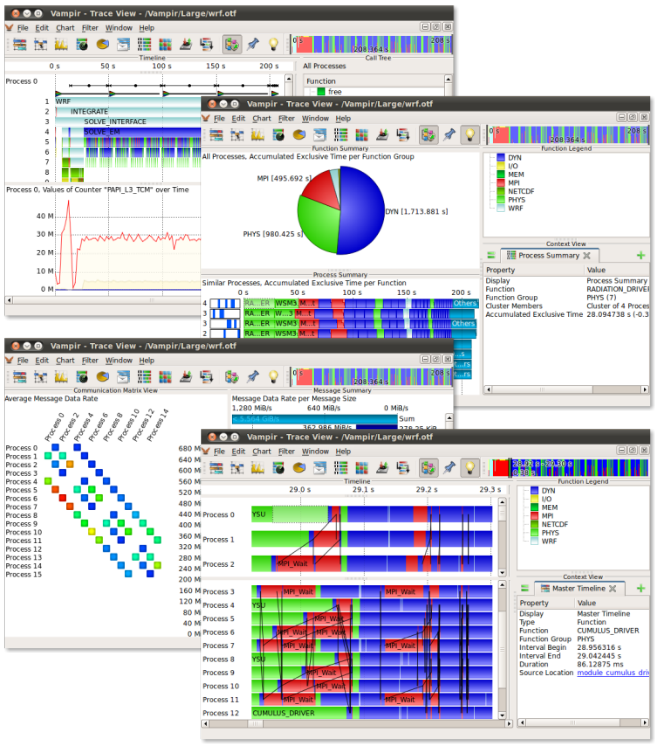

Sunway TaihuLight


| 0.28 PFLOP | 10.5 PFLOP | 93 PFLOP | 1000+ PFLOP |
|---|---|---|---|
| 131,072 PE | 705,024 PE | 10,649,600 PE | 100M+ PE |

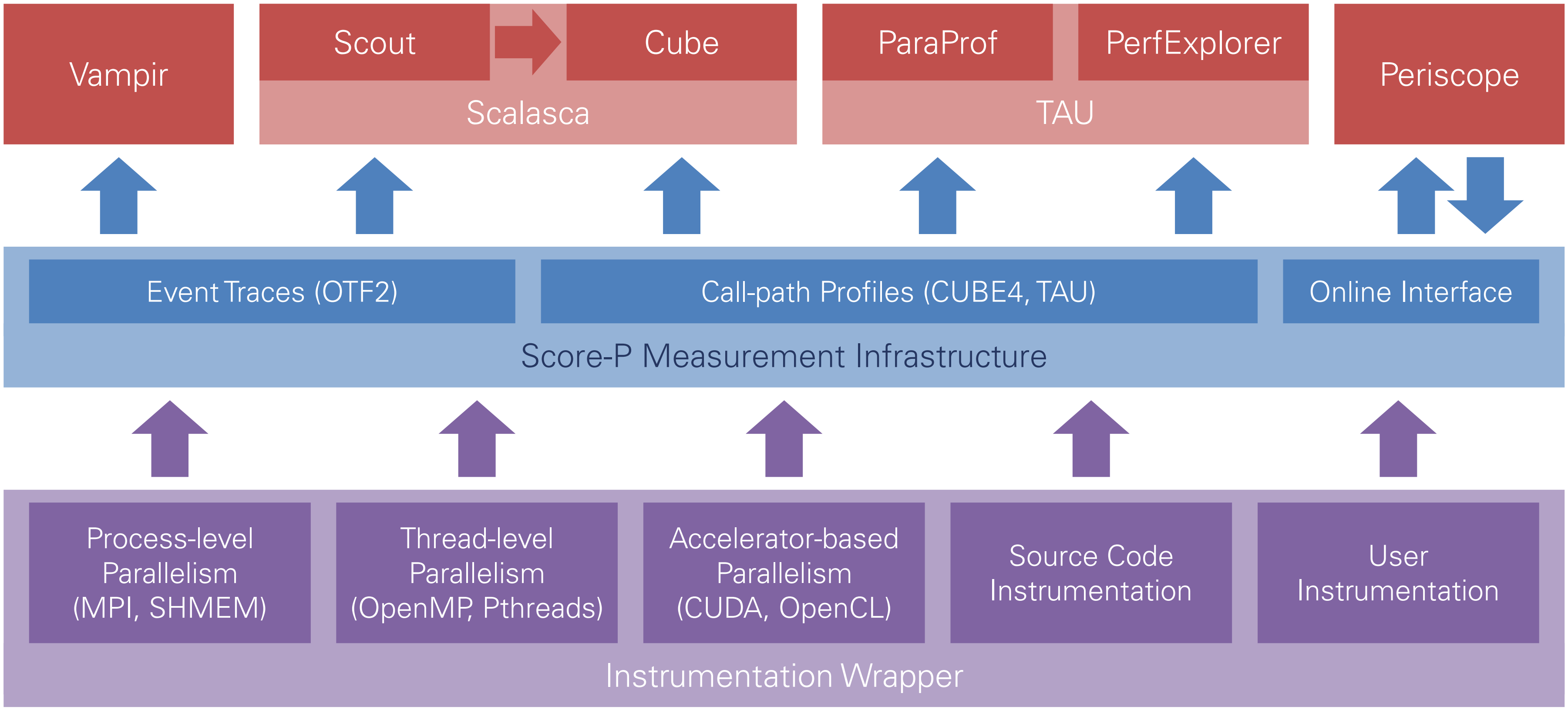
# Parallization – Ideal vs. Reality



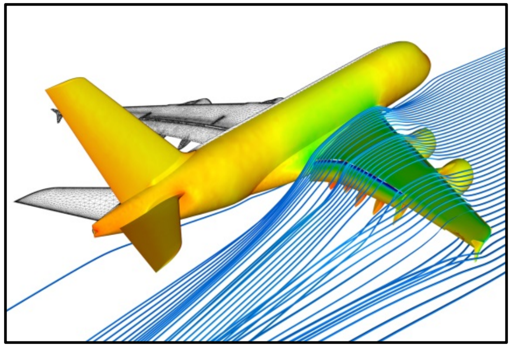

## Performance Analysis Tools



# Tool Workflow: Score-P, OTF2 and Analyzers



# Performance Analysis Workflow



Application

Measurement Tool

### Three Key Challenges

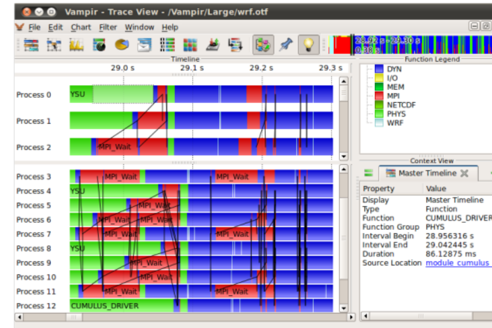
(1) Number of trace files limits scalability

(2) Huge amounts of trace data overwhelm file systems and analyzers

(3) Measurement bias due to intermediate memory buffer flushes

**Solution**

In-memory Event Tracing




Analysis

Analysis Tool

File System

# Concepts for In-memory Event Tracing

Complete Trace Data

Highly Efficient Encoding → Reduce memory allocation

Compact Trace Data

Dynamic Function Filtering → Filter highly frequent function calls

Filtered Trace Data

Event Reduction → Gradually reduce stored events

Memory Buffer

Reduced Trace Data

# Event Reduction

## Requirements

- Reduce number of stored events when memory buffer is exhausted
- Introduce minimal overhead
- Depend only on information extractable directly from events

## Comparison criteria

- Quality of remaining information
  - Is it still possible to understand the application behavior?
  - Is it still possible to identify performance issues?
- Size of single reduction steps

# Event Reduction

## (1) Reduction by Order of Occurrence

- Define time interval [t1,t2] with either t1 or t2 fixed
- Time interval contains complete information; none outside
- Small reduction steps (events)

## (2) Reduction by Event Class

- Sort events by class (functions, parallel library, performance metrics)
- Complete information for remaining event classes; none for others
- Large reductions steps (complete event classes)

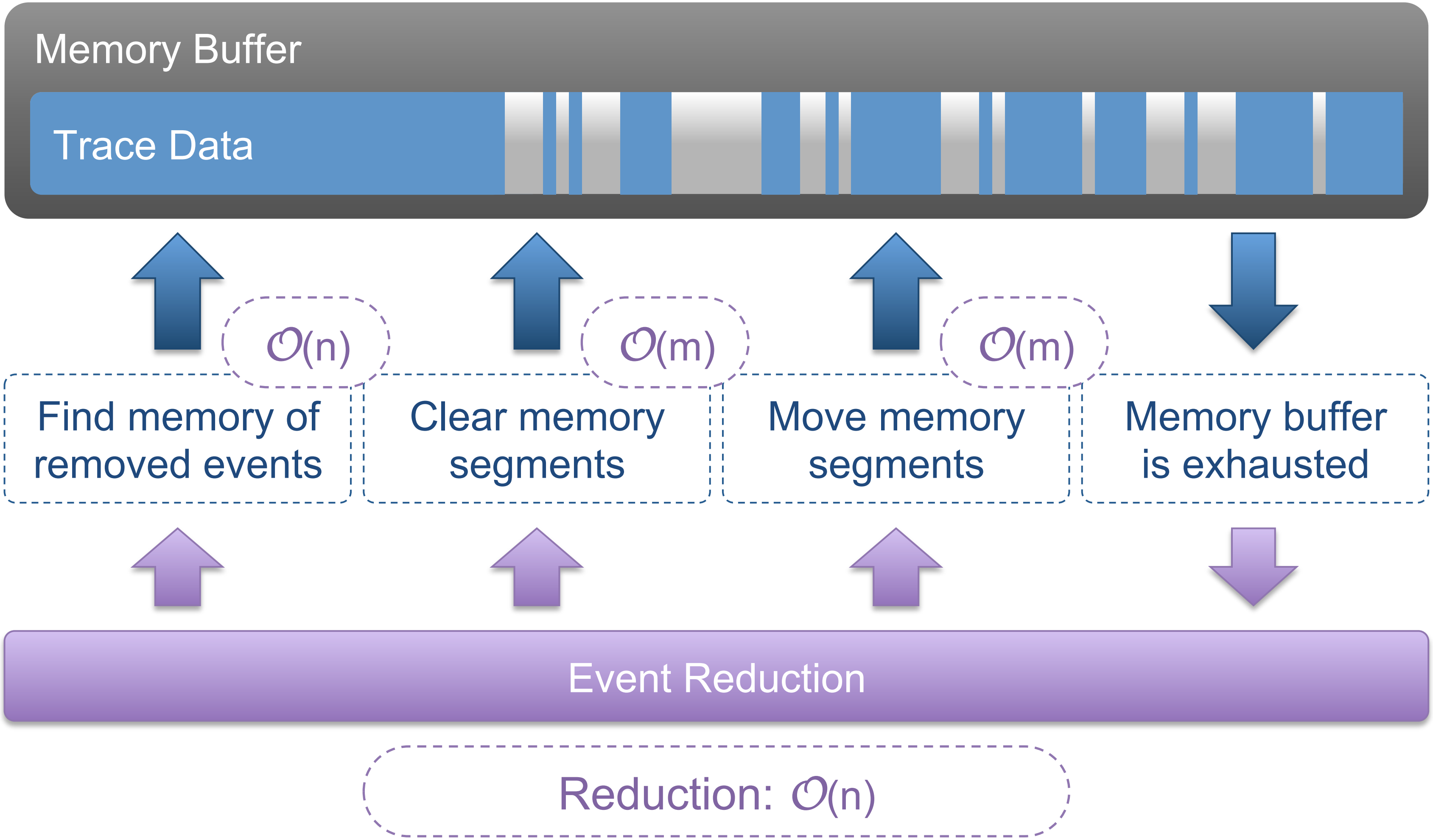
## (3) Reduction by Calling Depth

- Sort events by calling depth
- Overall information detail is reduced
- Depends on call stack distribution of events

## (4) Reduction by Function Duration

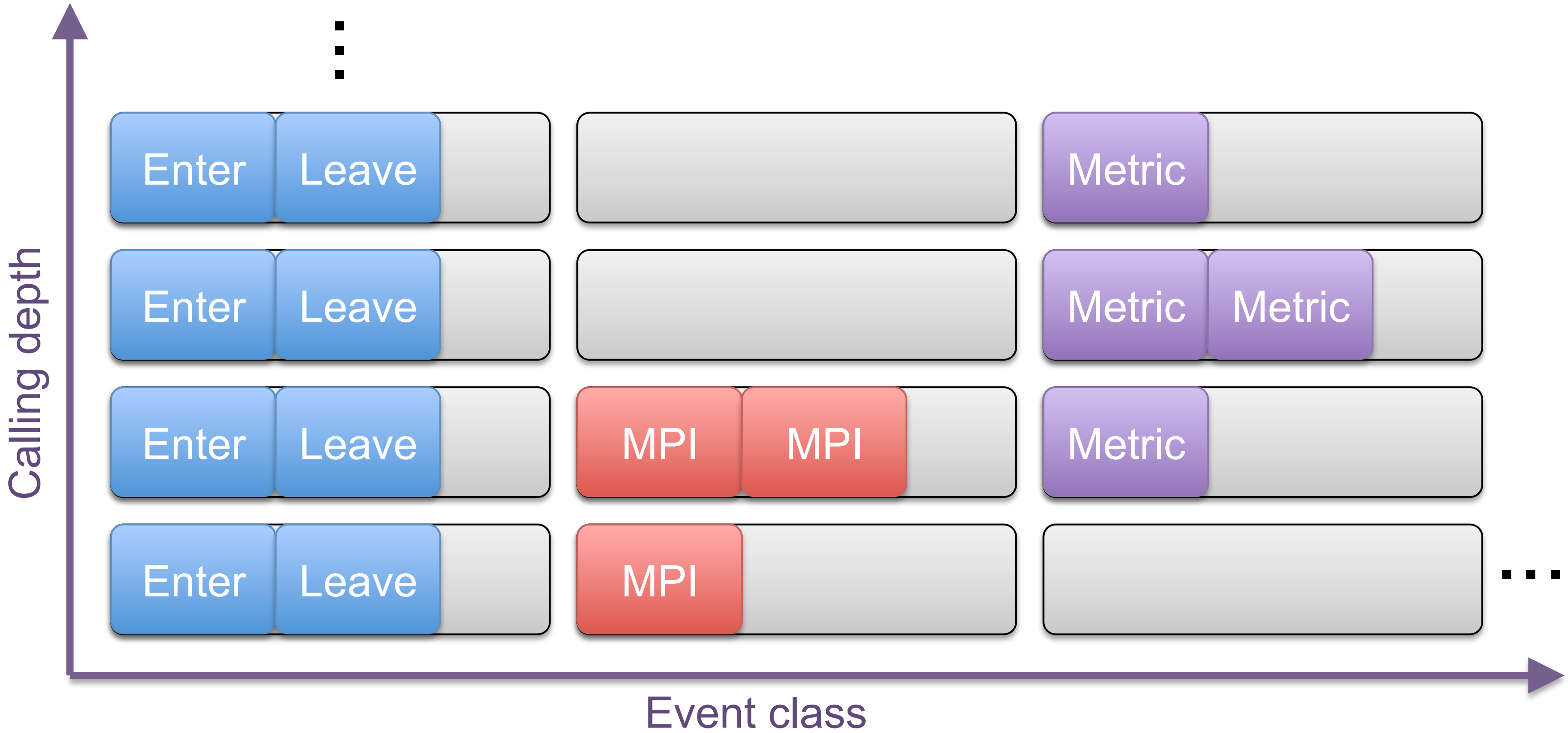
- Sort functions (enter/leave) by duration
- Overall information detail is reduced
- Depends on distribution of events with regard to function duration

# Event Reduction: Flat Continuous Event Representation

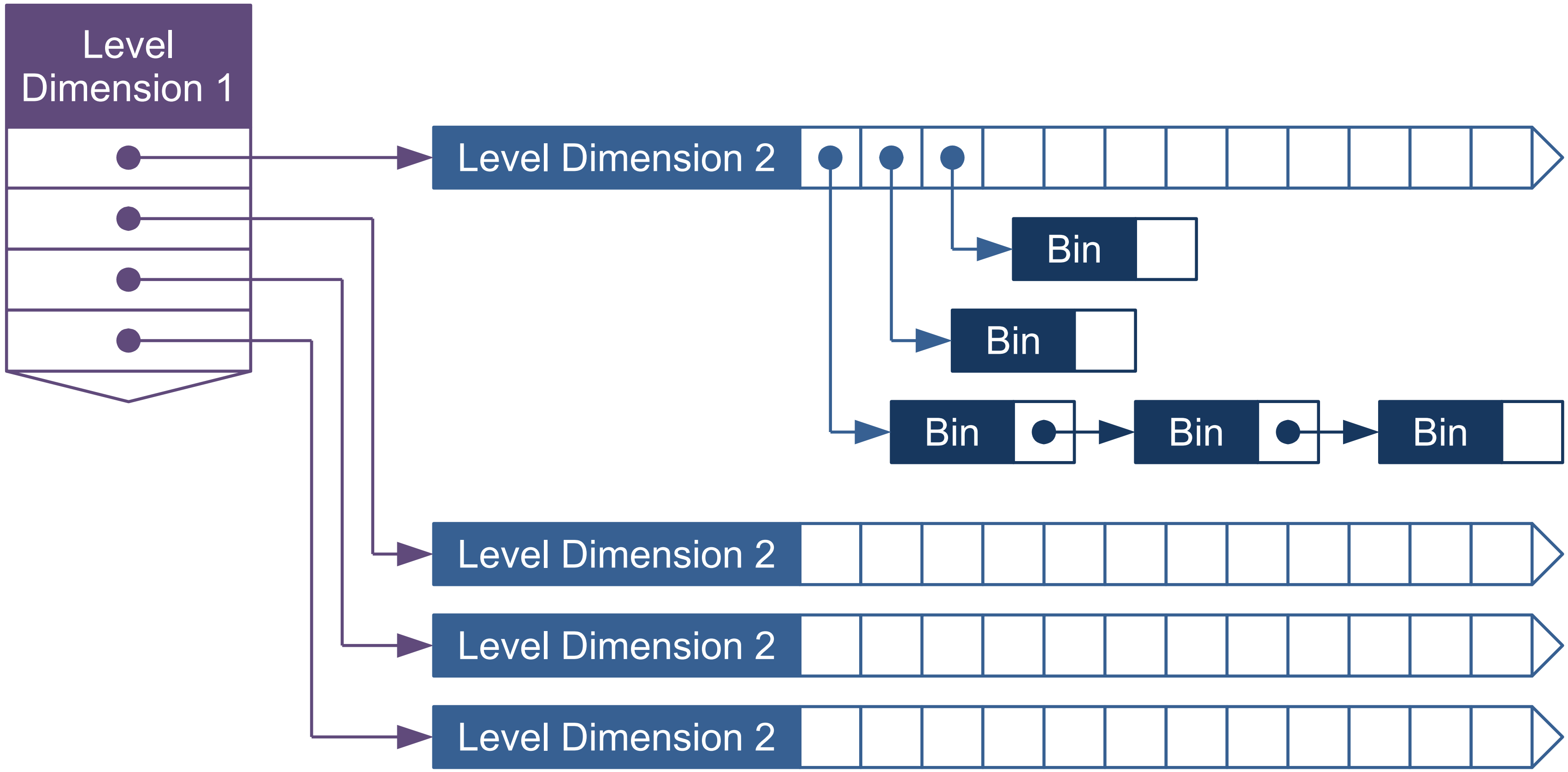


# Hierarchical Event Representation

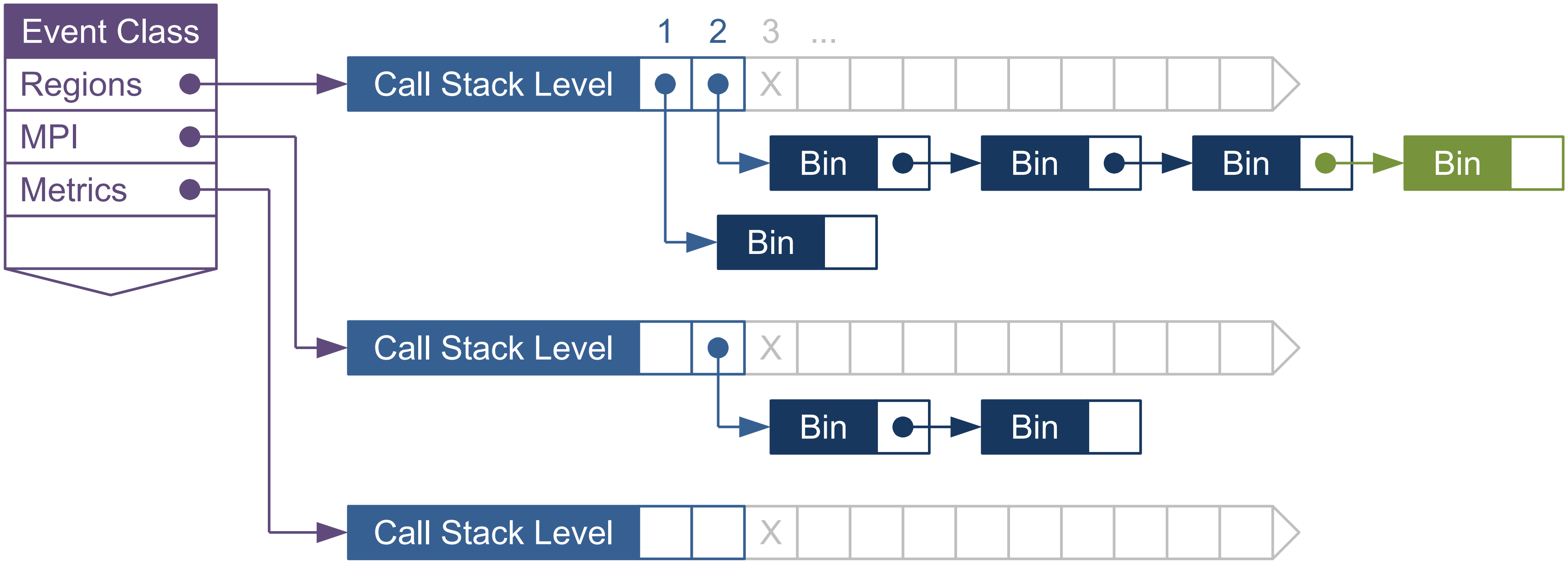
## Hierarchical Event Representation



# The Hierarchical Memory Buffer



# The Hierarchical Memory Buffer




$\mathcal{O}(1)$ — Find memory of removed events

$\mathcal{O}(b)$ — Revoke according memory bins

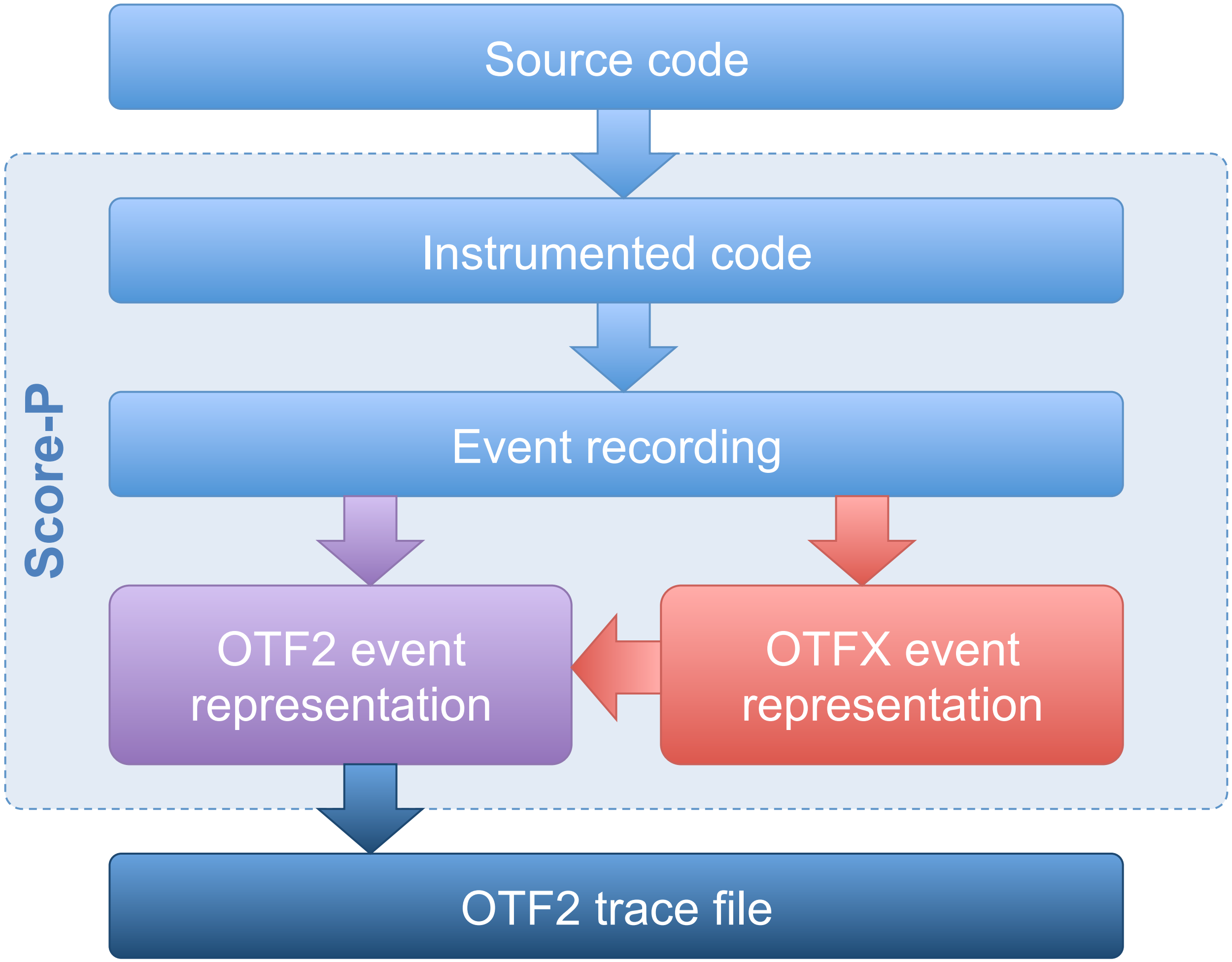
$\mathcal{O}(1)$ — Distribute memory bin

Memory buffer is exhausted

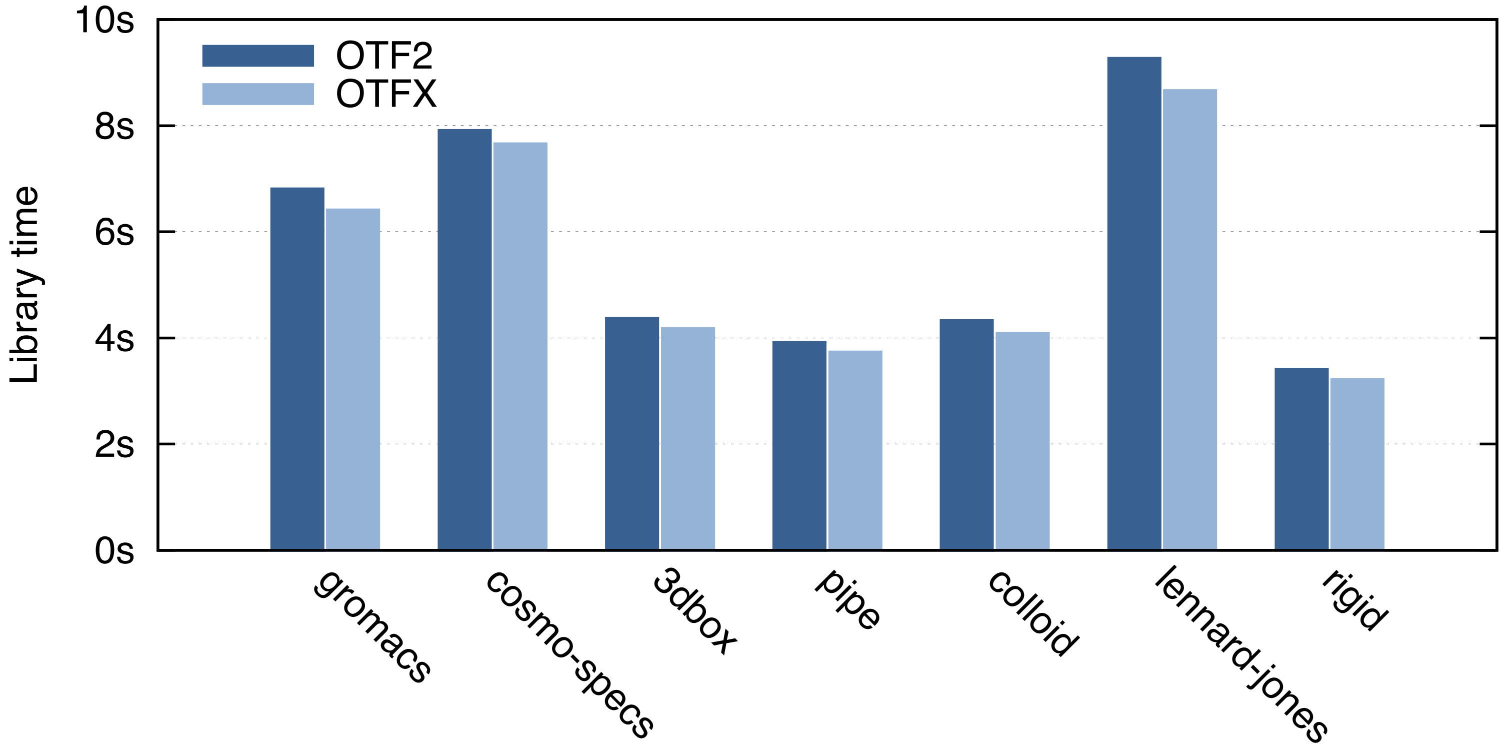
Event Reduction

Reduction: $\mathcal{O}(b)$

# Prototype Workflow with OTFX



# Evaluation: Runtime Overhead




- Trace replay to ensure equal input data for both libraries
- In average 5.1% faster than OTF2
- Library time of OTFX accounts for 7.8% of overall runtime

# Evaluation: Trace Sizes

| Application | Trace size (per process) | | | |
|---|---|---|---|---|
| | OTF2 | OTFX | +Filter | MPI-only |
| gromacs | 1.7 GB | 603 MB | 127 MB | 9.8 MB |
| cosmo-specs+fd4 | 1.5 GB | 514 MB | 21 MB | 80 KB |
| 3dbox | 919 MB | 297 MB | 116 MB | 8.8 MB |
| pipe | 817 MB | 267 MB | 88 MB | 8.5 MB |
| colloid | 900 MB | 266 MB | 40 MB | 12 MB |
| lennard-jones | 1.8 GB | 546 MB | 4.1 MB | 690 kB |
| rigid | 709 MB | 203 MB | 23 MB | 680 kB |

- OTFX compression results in 2.8x - 3.5x smaller traces
- Duration filter reduces trace to 0.2% - 12.6% of original size
- For gromacs and nek5000 (3dbox, pipe) event reduction is triggered

# Evaluation: Analysis

# Conclusion

- Tracing long-running applications encounters three critical challenges
  - Data volumes
  - Application slow down
  - Measurement bias
- In-memory event tracing workflow with OTFX
- Hierarchical memory buffer
- In-memory event tracing remarkably reduces trace size, application slow down and measurement bias