UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

FIB

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Heuristic-Based Merging of HPC Traces

Extending Hardware Counter Coverage for Performance Prediction

Júlia Orteu Aubach

Fabio Banchelli, Marc Clascà and Marta Garcia-Gasulla

Contents

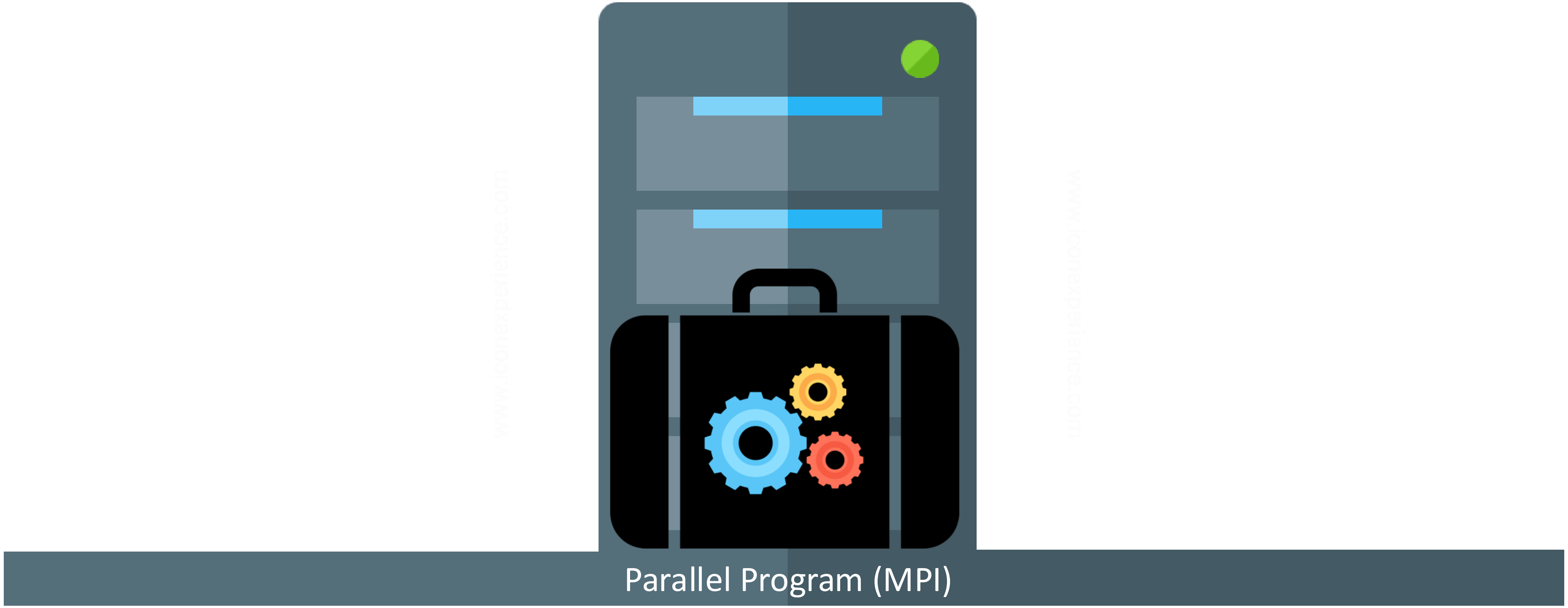
Context

High Performance Computing (HPC)

Context

Supercomputer

Context



Program Execution

Context

Extrae tool

- Instrumentation without code changes or recompilation
- MPI Applications
- C, C++ and Fortran

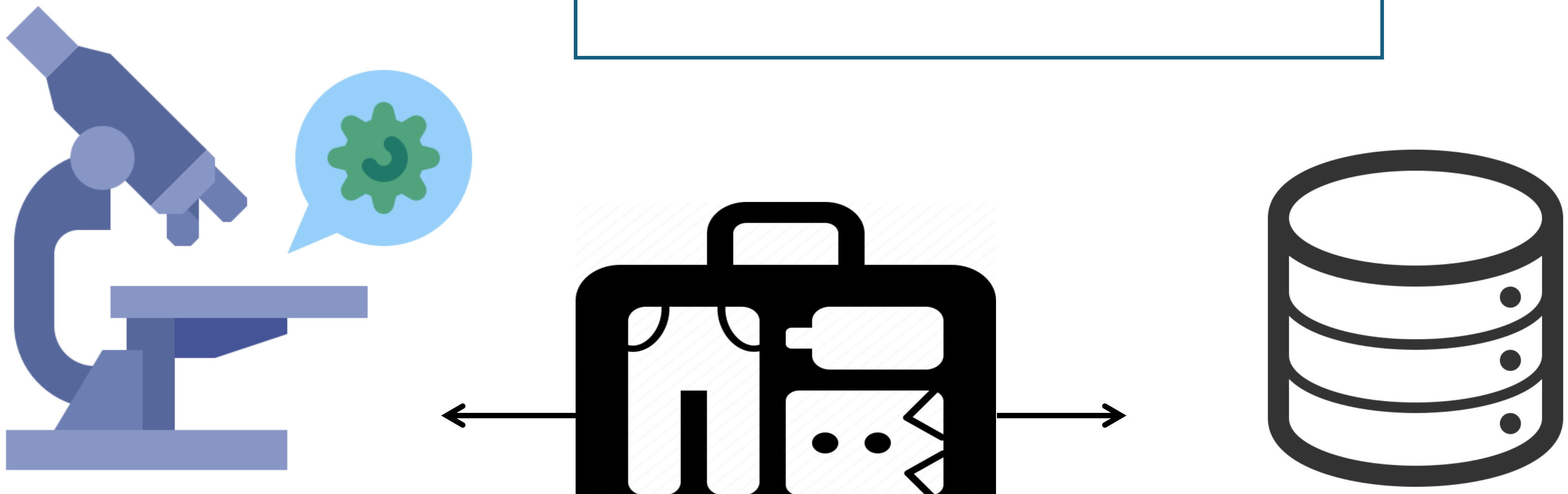
Parallel Program (MPI)

Program Execution

Context

Paraver tool

- Paraver execution trace viewer in timeline format
- Automatic calculation of user-defined metrics

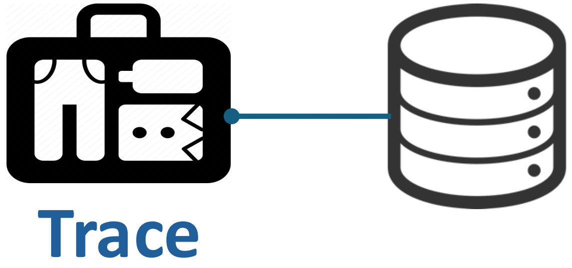


Performance Analysis

Extrae **Trace**

Large Volume of Data

Context



In previous work the objective was to develop and train a machine learning (ML) model using simple parallel programs (benchmarks and kernels) and hardware counters to **predict the performance of unseen applications on an HPC machine.**

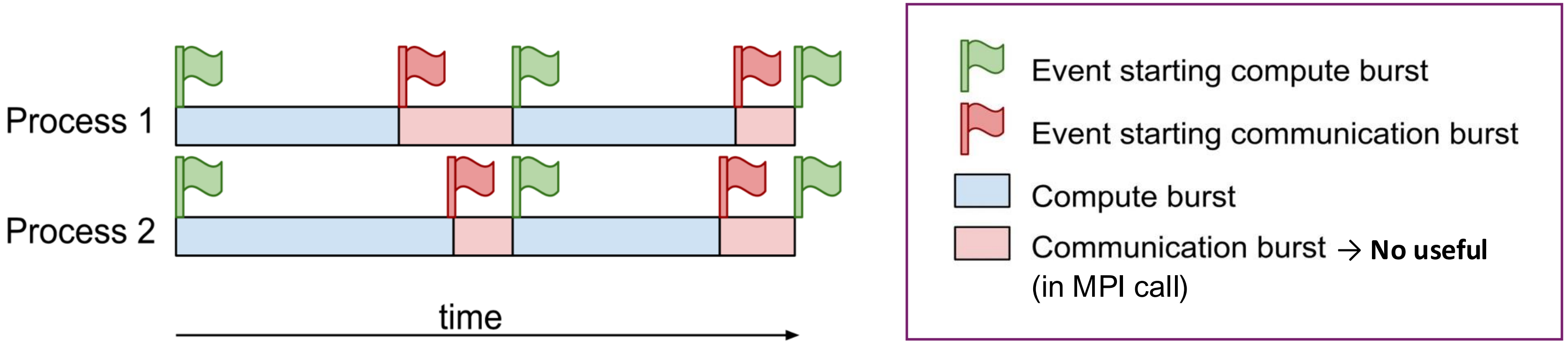
Orteu, J., Clascà, M., Labarta, J., Jennings, E., Andersson, S., Garcia-Gasulla, M. (2025).
A Framework and Methodology for Performance Prediction of HPC Workloads.
Parallel Processing and Applied Mathematics. PPAM 2024. Lecture Notes in Computer Science, vol 15581. Springer, Cham.
https://doi.org/10.1007/978-3-031-85703-4_8

Context

What type of data?

Bursts

We define a burst as the time interval between two successive events in a process

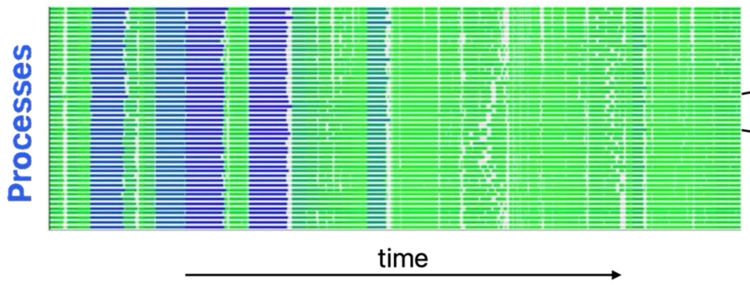


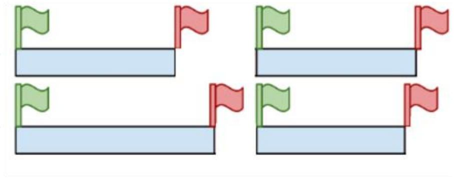
Context

What type of data?

Compute Bursts

Example application trace



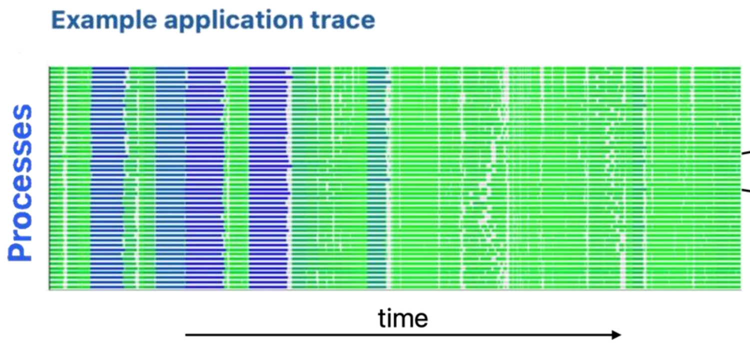


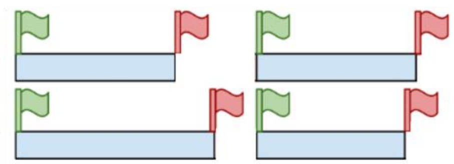
From each burst we can extract hardware counters...

Context

What type of data?

Compute Bursts





From each burst we can extract hardware counters…

Performance Application Programming Interface (**PAPI**) provides access to hardware counters

PAPI_TOT_INS
Total Instructions

PAPI_TOT_CYC
Total Cycles

PAPI_L1_DCM
L1 Data Cache Misses

PAPI_VEC_DP
Vector Double Precision

PAPI_BR_INS
Branch Instructions

PAPI_FP_OPS
Floating Point Operations

Context

Data Format

We extract a tabular data set for each execution trace.

Row = Bursts
Cols = Trace features
- Timeline information
- MPI contextual information
- Hardware Performance counters

TaskId	Begin_Time	Duration	INS	End_Time	Line	d_IPC	x_PAPI_L1_DCM	x_PAPI_L2_DCM	...
1	7254812180	71362947	100059606	7.326175e+09	630.0	5.128201	223733.0	185399.0	...
1	7326181549	5026	2158	7.326187e+09	1548.0	0.222245	133.0	89.0	...
1	7326189542	13978	11780	7.326204e+09	1552.0	0.548392	338.0	178.0	...
1	7340619255	6277	3612	7.340626e+09	1668.0	0.481793	167.0	75.0	...
1	7340635605	1568384	16459717	7.342204e+09	1680.0	5.168772	1098.0	25.0	...
...	...	...	...	...	...	...	...	...	...

Context

What type of data?

🚧 **The Limitation**

- Hundreds of available counters, but only **4-8 can be measured simultaneously** due to architectural constraints
- This creates an **"n-counter ceiling"** for our ML models

Performance Application Programming Interface (**PAPI**) provides access to hardware counters

PAPI_TOT_INS Total Instructions	**PAPI_TOT_CYC** Total Cycles	**PAPI_L1_DCM** L1 Data Cache Misses
PAPI_VEC_DP Vector Double Precision	**PAPI_BR_INS** Branch Instructions	**PAPI_FP_OPS** Floating Point Operations

Context

Existing solutions and their limitations

Fixed Counter Selection

❌ Limits feature diversity

Hardware Multiplexing

❌ Introduces timing noise

❌ Intermittent sampling

Objectives

Objectives

Research Objectives

1. Overcome **the n-counter ceiling** in HPC performance modeling.

2. Maintain **burst-level fidelity** required for detailed analysis.

3. Create synthetic traces compatible with **existing tools** (Paraver/Extrae).

4. Validate methodology across **diverse HPC applications.**

Methodology

Methodology Overview

Benchmark Applications

Application	Domain	Description
SOD2D	Computational Fluid Dynamics	Spectral high-Order code for solving partial differential equations, primarily used for fluid dynamics and wave propagation problems
SeisSol	Seismic Simulation	High-performance seismic wave simulation software for earthquake modeling and ground motion prediction
Stream	Memory Benchmark	Memory bandwidth benchmark measuring performance of four vector operations: Copy, Scale, Add, and Triad
Alya Solver	Computational Mechanics	Mini-app from HPC mechanics application featuring very fine-grained parallelism for computational mechanics problems
Lulesh	Hydrodynamics	Benchmark tool for shock wave simulations in fluid dynamics, focusing on efficient energy calculations for nuclear fission explosions

Methodology Overview

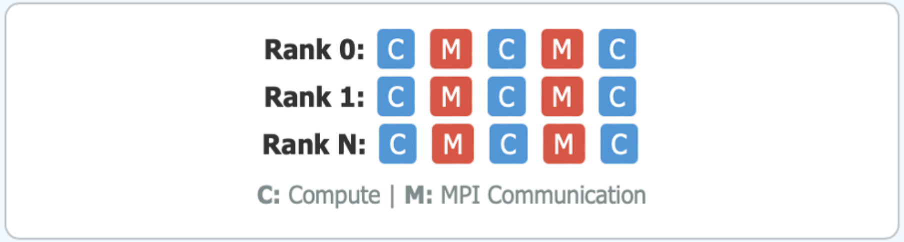
Observed Communication Patterns Across Applications

1. Processes with Identical Communication Behavior

Characteristics:

- ▶ Consistent structure between different executions
- ▶ Each rank may have different burst counts, but this remains constant across runs
- ▶ Enables direct, position-based burst matching

Examples: Stream, Alya, Lulesh



Methodology Overview

Observed Communication Patterns Across Applications

2. Processes with Structural Variations

Characteristics:

- ▶ Different MPI communication patterns between executions
- ▶ Variable number of bursts per run (moderate variations)
- ▶ Some recurring structures can be identified for pattern-based matching
- ▶ Less irregular than highly complex applications, but not fully deterministic

Example: SOD2D

Execution 1 - Rank 0: C B C A C
Execution 2 - Rank 0: C B C F C A

C: Compute | **B:** Barrier | **A:** Allreduce | **F:** Comm_Free
Note: Different MPI calls and burst counts between runs

Methodology Overview

Observed Communication Patterns Across Applications

3. Highly Irregular Traces

Characteristics:

- ▶ Complex and heterogeneous communication behavior
- ▶ Significant structural irregularities
- ▶ Requires sophisticated matching with structural constraints

Rank 0: C I C M C I
Rank 1: I I C M C
Rank N: C I I I C M C

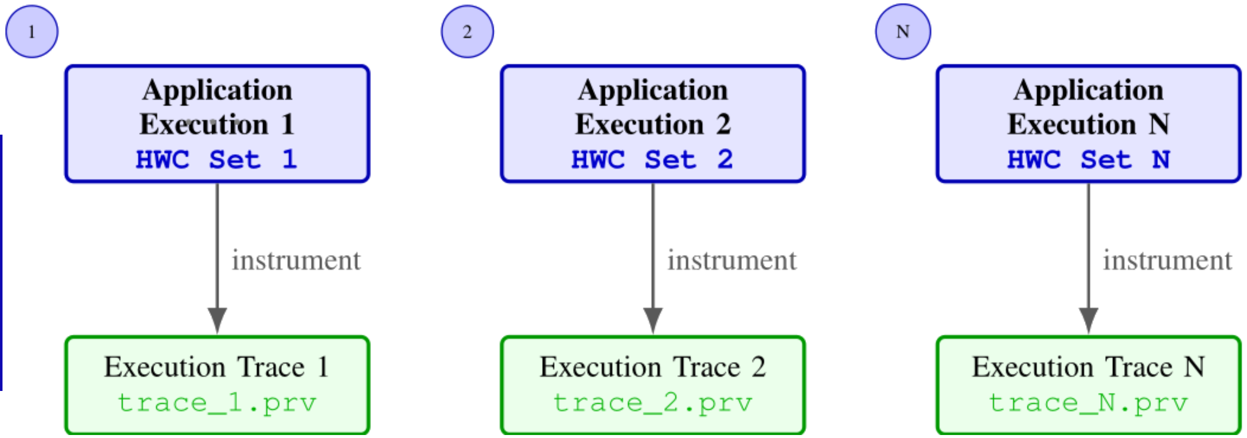
C: Compute | **M:** MPI Collective | **I:** MPI Irregular (ISEND/IRECV/TEST)

Example: SeisSol

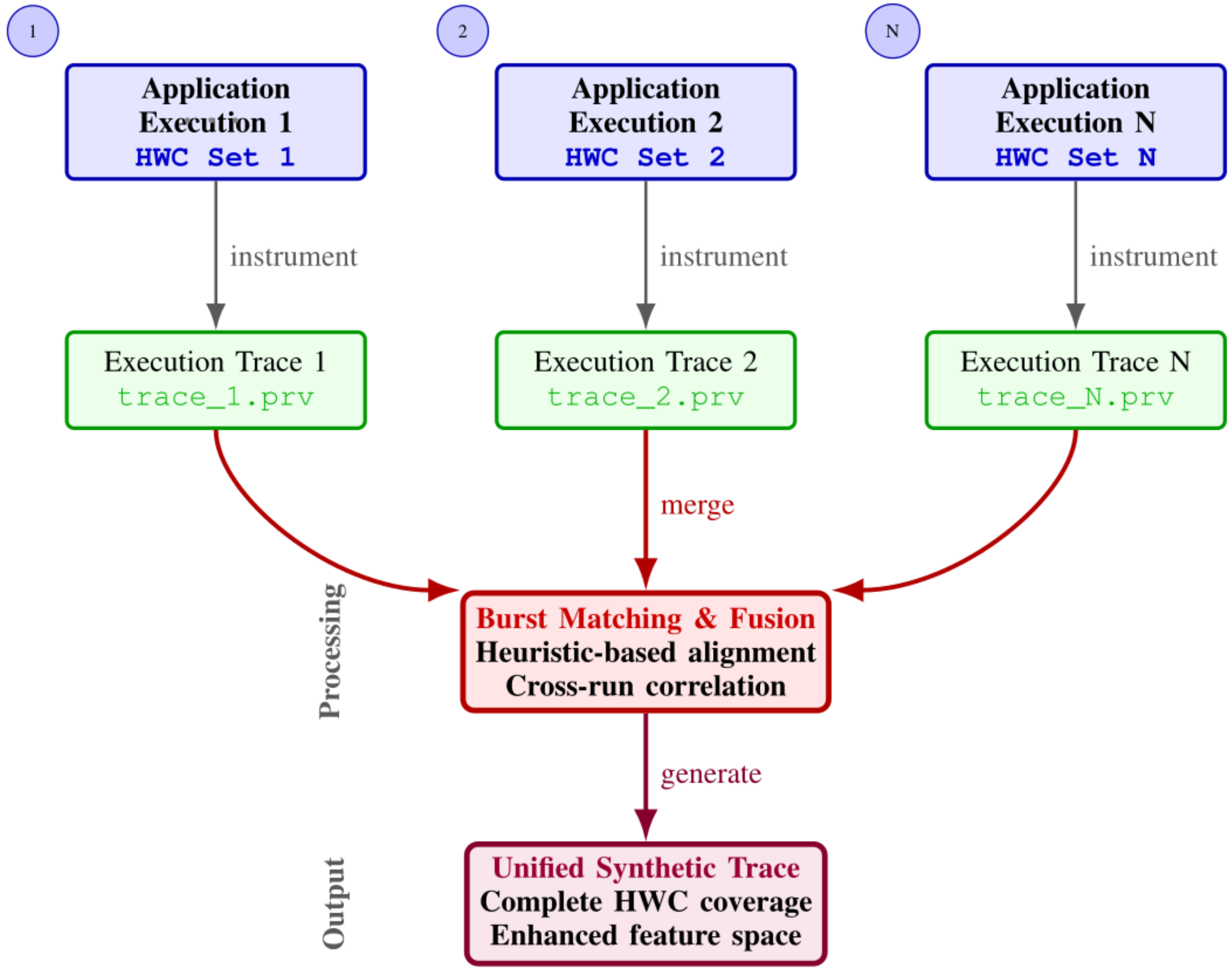
These observed patterns guided the design of our **two-stage matching algorithm**, which progressively increases complexity based on structural similarity detection.

Methodology Overview

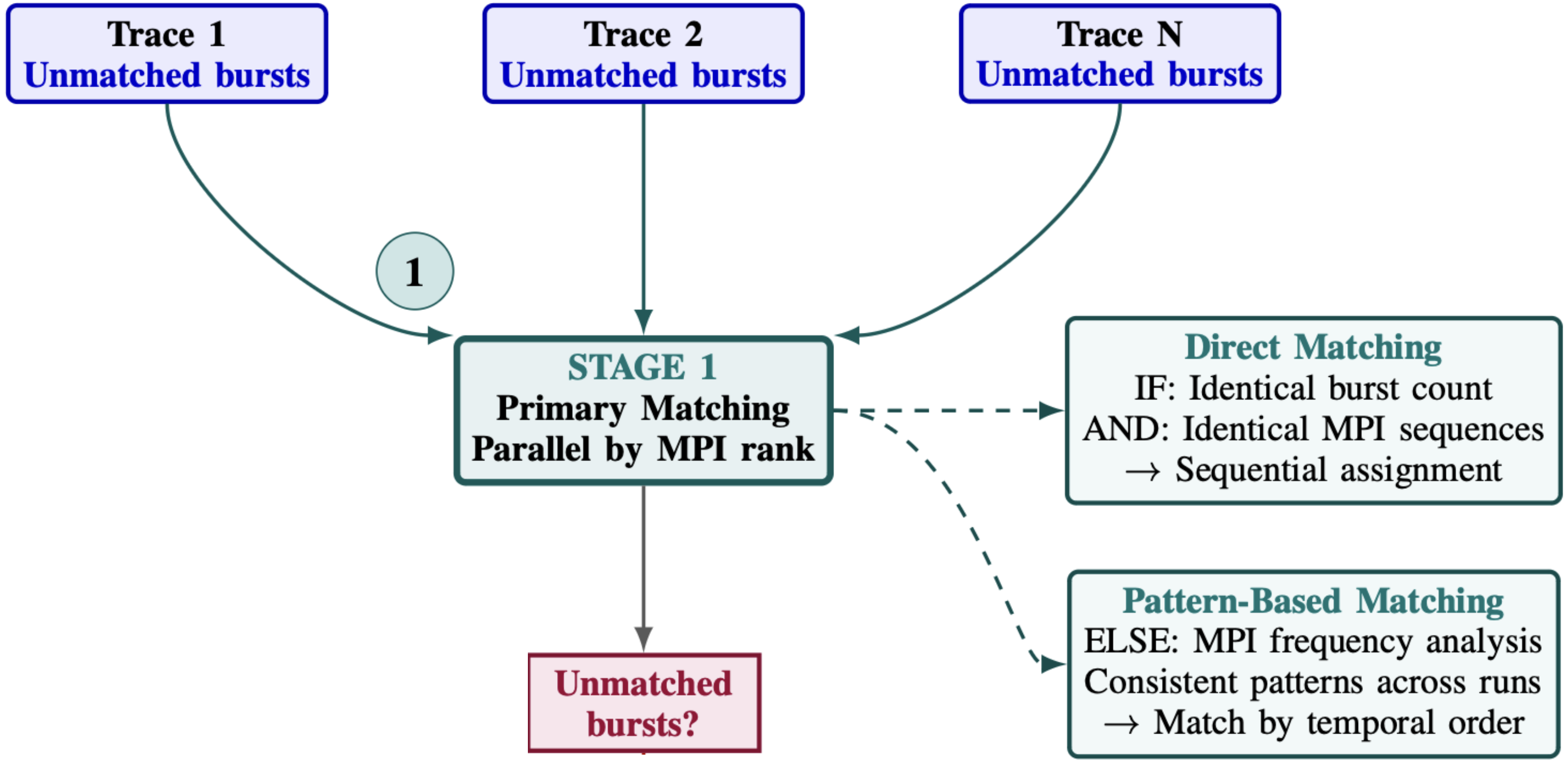
The methodology scales to accommodate **multiple executions with diverse counter configurations**, limited only by the available hardware counter combinations



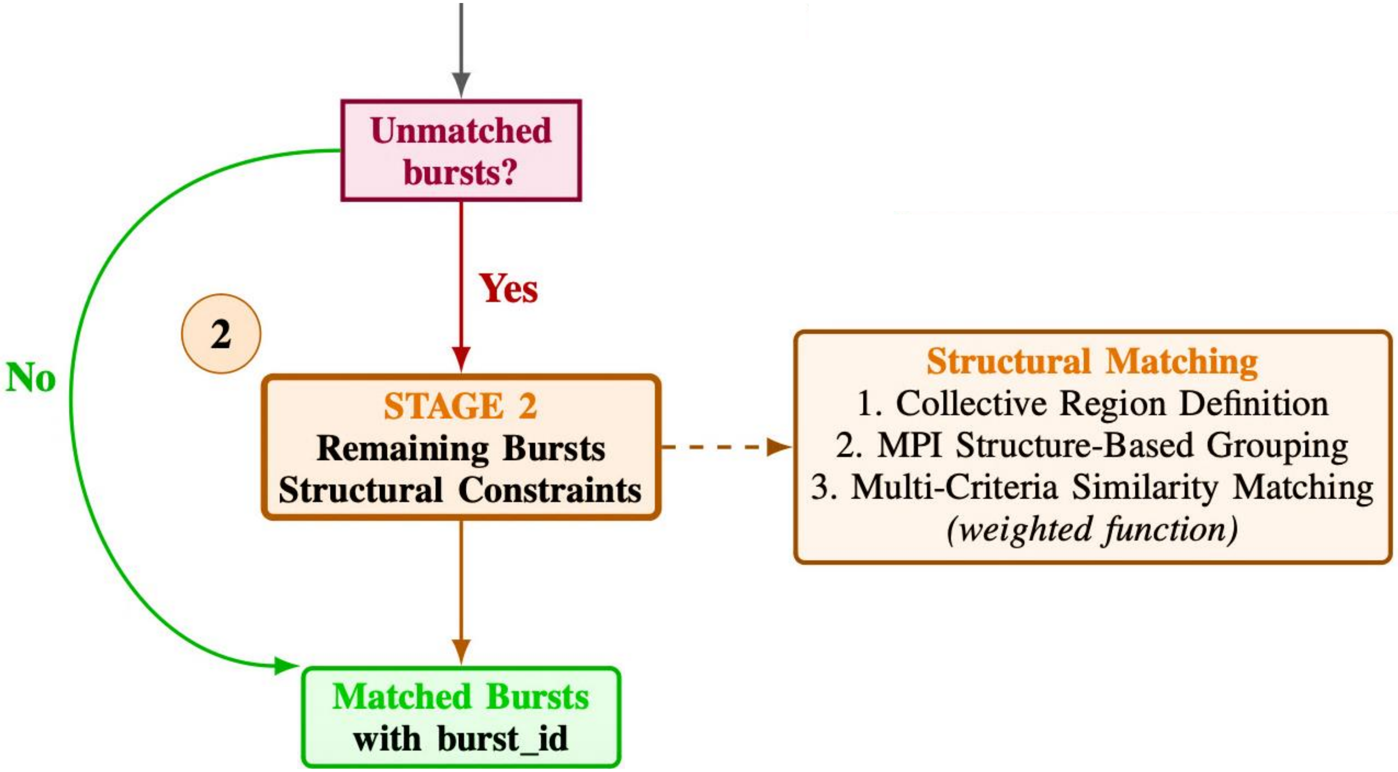
Methodology Overview



Methodology Overview



Methodology Overview



Methodology Overview

Structural Matching

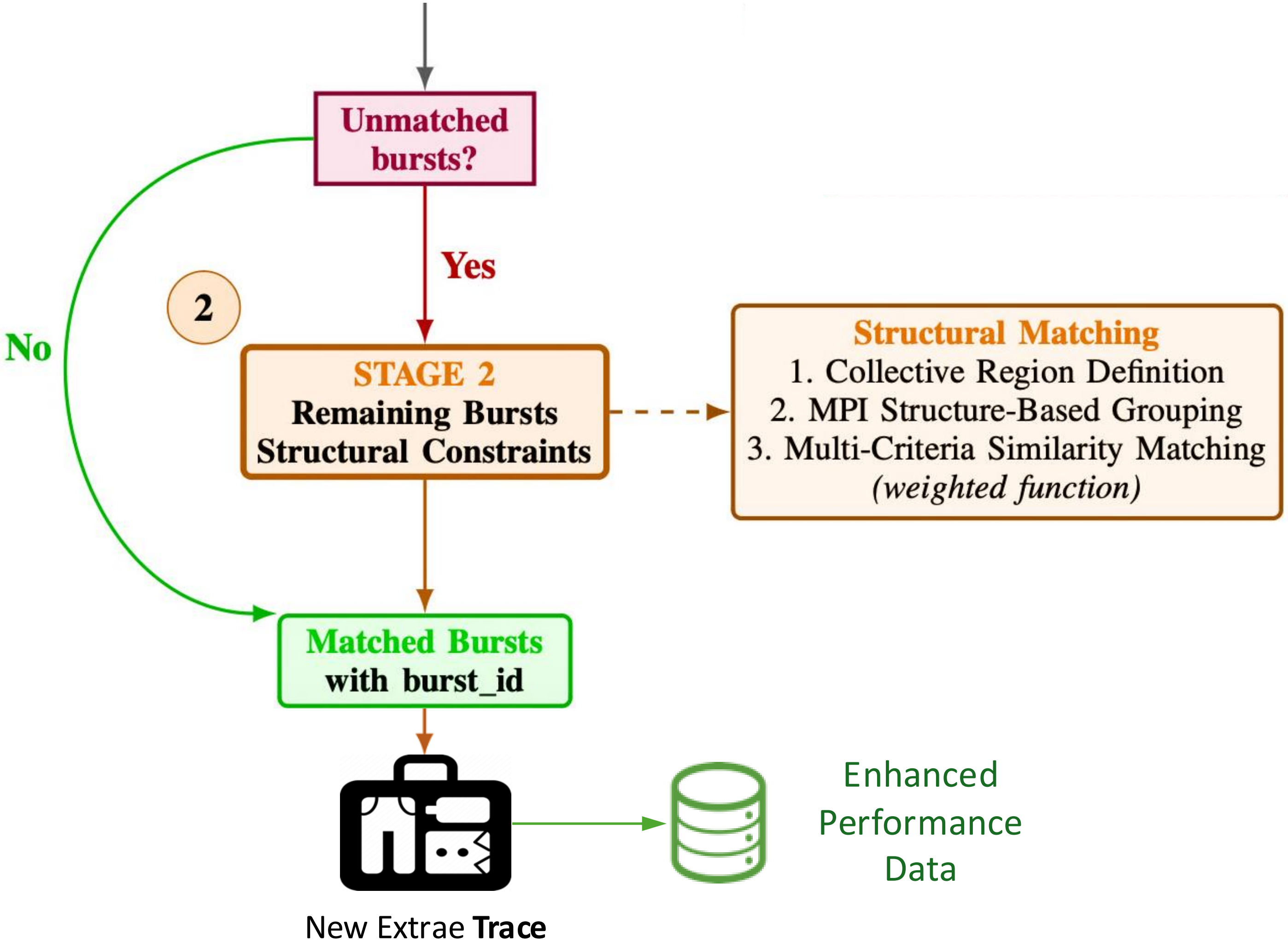
1. Collective Region Definition
2. MPI Structure-Based Grouping
3. Multi-Criteria Similarity Matching *(weighted function)*

$$S = 0.6 \times D_{temporal} + 0.2 \times D_{size} + 0.2 \times D_{partner}$$

Temporal Position	Communication Size	Communication Partner
Relative position within collective regions	Message sizes for MPI operations	Rank identifiers for point-to-point
Weight: 60%	**Weight: 20%**	**Weight: 20%**

Matching Threshold: $S < 0.3$ for genuine similarity

Methodology Overview



Experimentation

Experimentation Results

MareNostrum5 Configuration

Execution Parameters

- **Processes:** 100-112 MPI processes
- **Repetitions:** 2-10 runs per configuration
- **Noise Reduction:** System isolation

Hardware & System

- **Platform:** Intel Sapphire Rapids CPUs
- **Frequency:** Fixed at 2 GHz
- **Allocation:** Exclusive node access

Experimentation Results

Three Different Hardware Counter Configurations

INS_MIX

Focus: Instruction mix & cache hierarchy

PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_LD_INS, PAPI_SR_INS, PAPI_BR_INS, PAPI_L3_TCM, PAPI_L1_DCM, PAPI_L2_DCM

OPS_SET

Focus: Floating-point operations & vectorization

PAPI_TOT_INS, PAPI_VEC_INS, PAPI_FP_INS, PAPI_FP_OPS, PAPI_DP_OPS, PAPI_SP_OPS, PAPI_VEC_SP, PAPI_VEC_DP

OPS_CYC

Focus: Computational performance

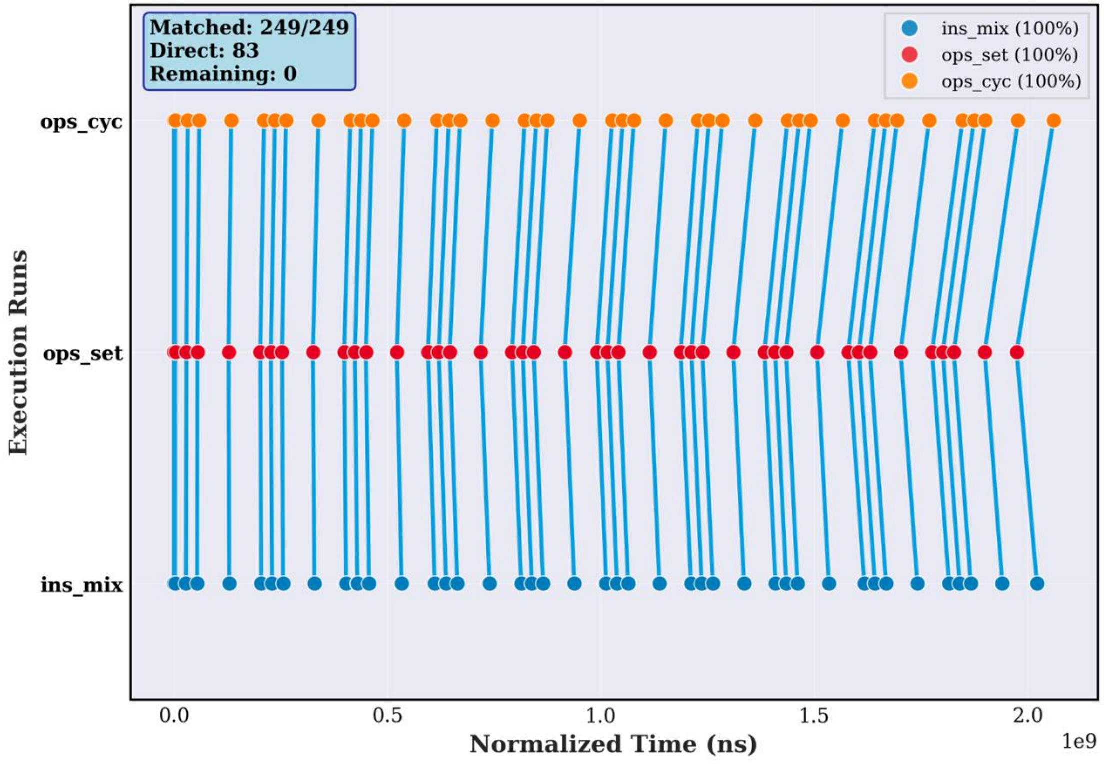
PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_VEC_DP, PAPI_VEC_SP, PAPI_DP_OPS

Experimentation Results

Applications with Identical Communication Behaviour

Experimentation Results

Applications with Identical Communication Behaviour



Stream application for TaskId 9

Blue lines connect corresponding bursts identified by the matching algorithm (Stage 1).

Experimentation Results

Applications with Identical Communication Behaviour

Stream

100%

9,296 bursts | 112 MPI ranks

Direct Matching

Memory bandwidth benchmark with identical burst patterns

Alya

100%

16,428 bursts | 48 MPI ranks

Direct Matching

Computational mechanics with deterministic structure

Lulesh

100%

43,077 bursts | 27 MPI ranks

Direct Matching

Hydrodynamics with consistent MPI patterns

Experimentation Results

Processes with Estructural Variations Between Executions

Execution 1 - Rank 0: C B C A C

Execution 2 - Rank 0: C B C F C A

C: Compute | **B:** Barrier | **A:** Allreduce | **F:** Comm_Free

Note: Different MPI calls and burst counts between runs

Experimentation Results

Processes with Estructural Variations Between Executions

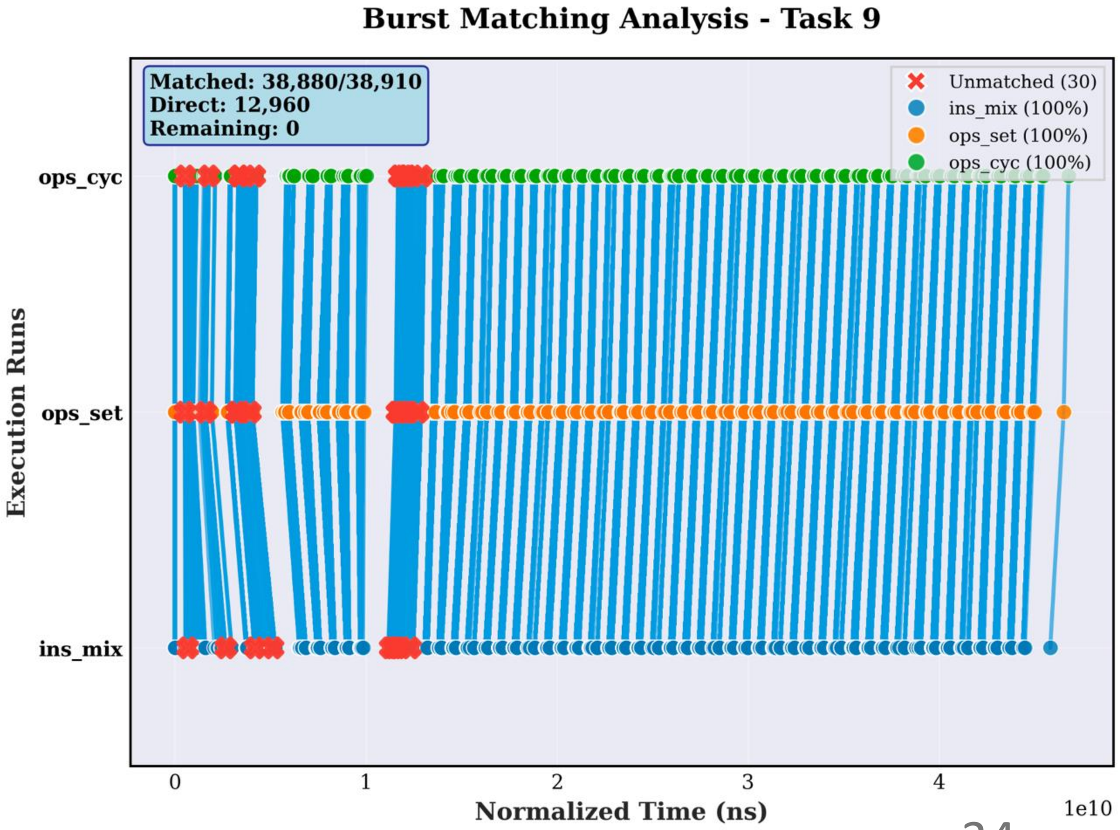
Execution 1 - Rank 0: C B C A C

Execution 2 - Rank 0: C B C F C A

→ 3 executions →

C: Compute | **B:** Barrier | **A:** Allreduce | **F:** Comm_Free

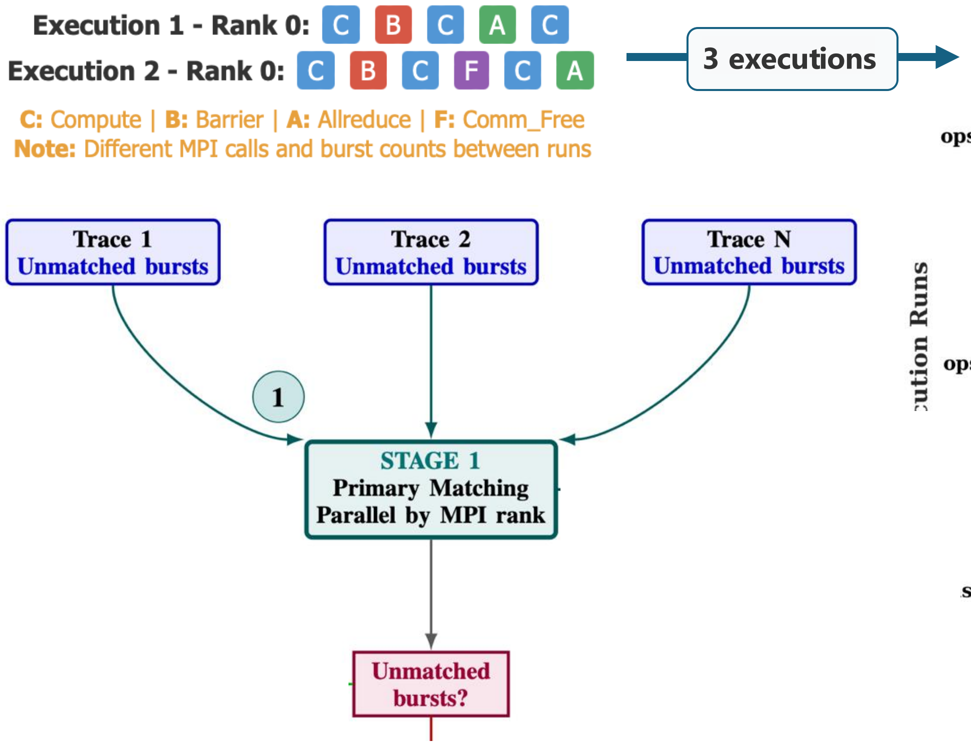
Note: Different MPI calls and burst counts between runs

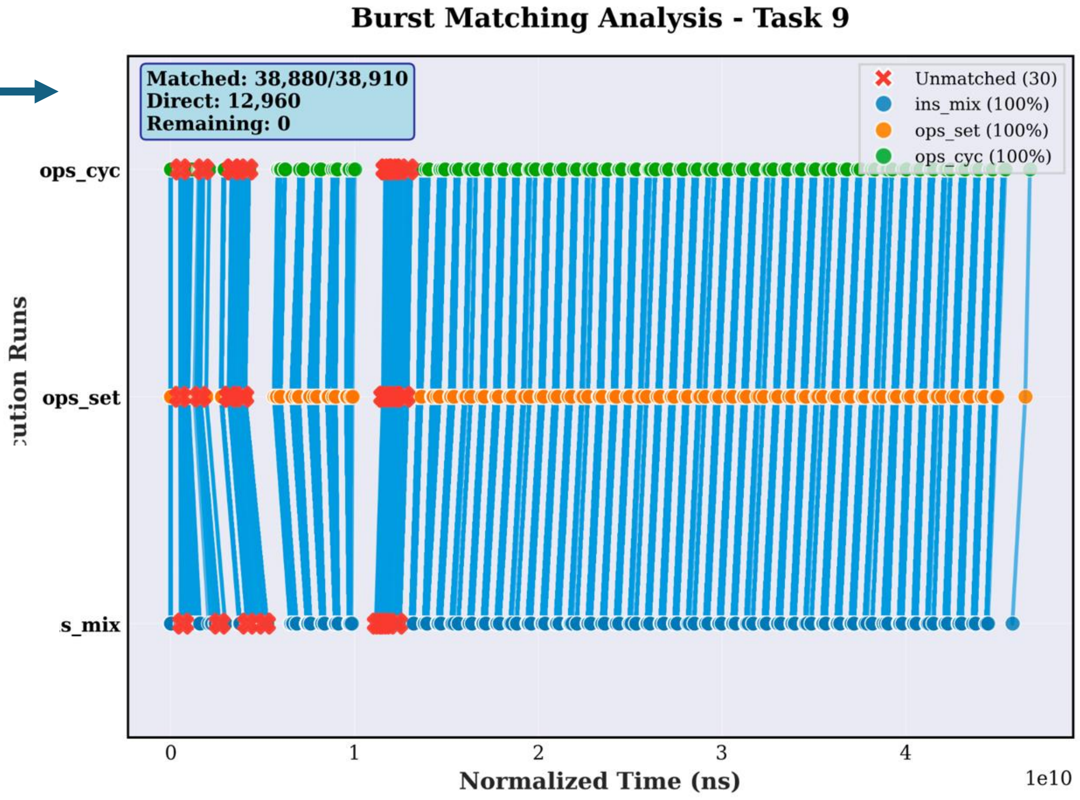


SOD2D application for TaskID 9

Experimentation Results

Processes with Estructural Variations Between Executions

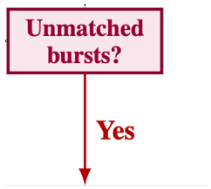


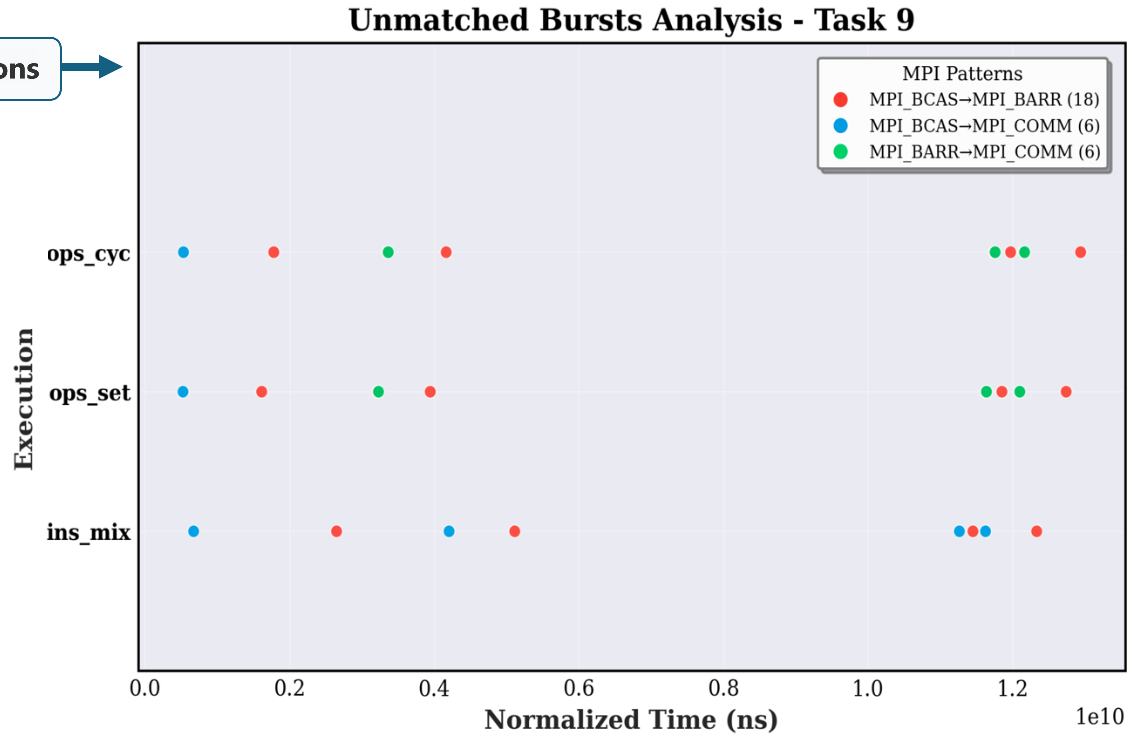


SOD2D application for TaskID 9

Experimentation Results

Processes with Estructural Variations Between Executions

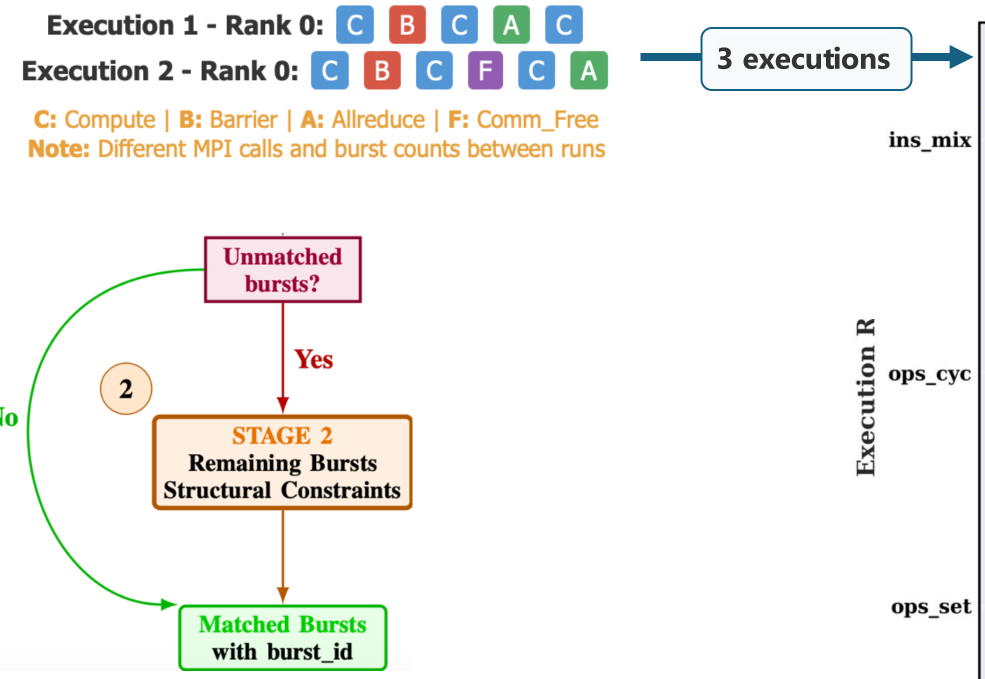


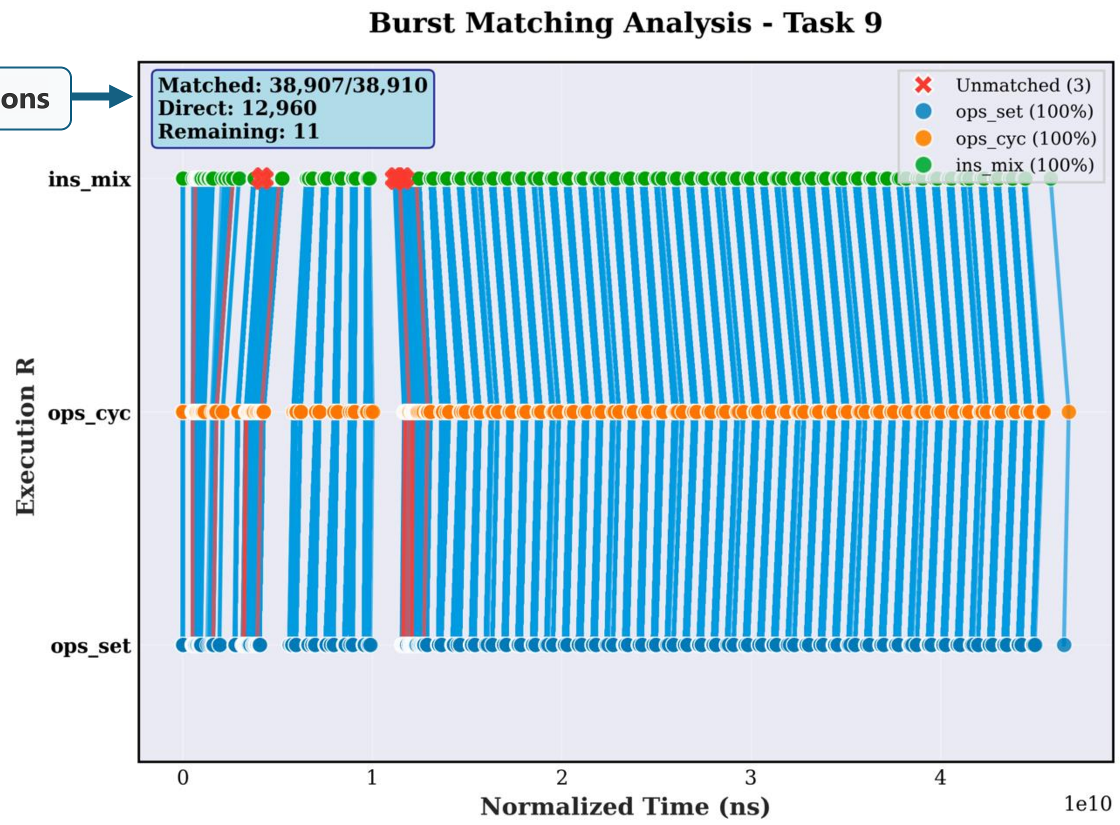


SOD2D application for TaskID 9

Experimentation Results

Processes with Estructural Variations Between Executions





SOD2D application for TaskID 9

Red lines connect corresponding bursts identified by the matching algorithm in Stage 2

Experimentation Results

Processes with Estructural Variations Between Executions

SOD2D

99.99%

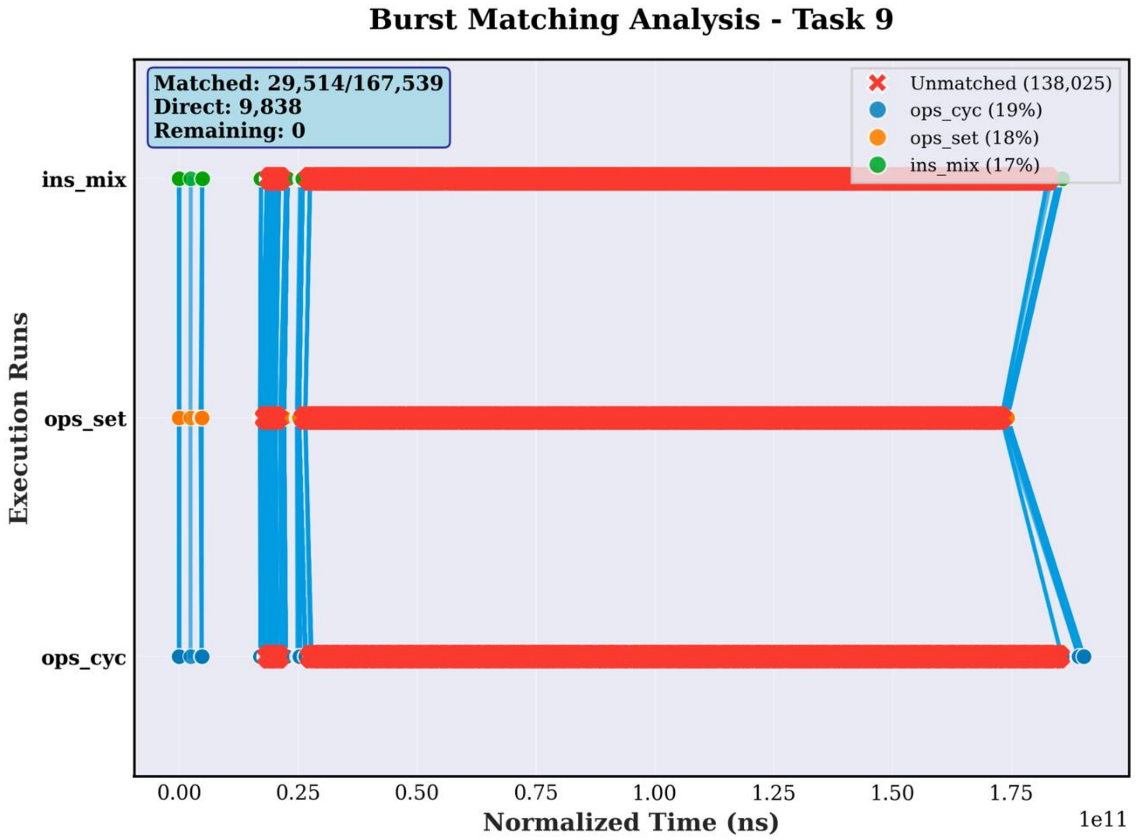
~1.5M bursts total

Pattern-Based

CFD with minor structural variations (0.02% difference)

Experimentation Results

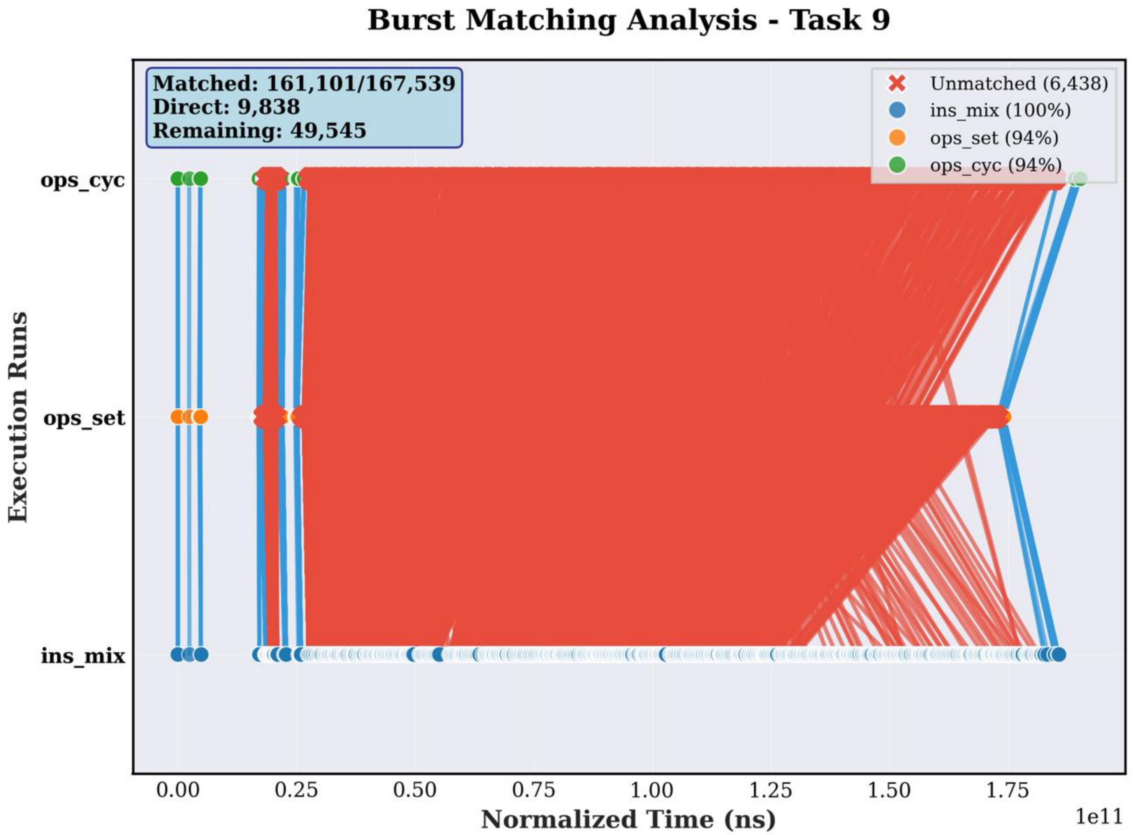
Highly Irregular Traces



SeisSol application for TaskID 9

Experimentation Results

Highly Irregular Traces



SeisSol application for TaskID 9

Experimentation Results

Highly Irregular Traces

SeisSol

90-99%

>5.5M bursts | 128 MPI patterns

Complex Matching

Seismic simulation with significant non-determinism

Experimentation Validation

Validation Framework

Apply the same heuristic matching algorithm to N traces with **identical counter sets** to quantify matching precision and establish baseline accuracy.

1. **Controlled Setup** — N executions with identical counter sets per application
2. **Algorithm Application** — Apply matching heuristics to identify burst correspondences
3. **Quality Assessment** — Calculate three validation metrics for matched bursts
4. **Statistical Analysis** — Correlation matrices, distributions, and visual validation

Experimentation Validation

3

Quality Assessment

Pearson Correlation

Correlation coefficient between base trace and matched traces

`r > 0.95 = High Accuracy`

Mean Absolute Error

Absolute deviation in original counter units

$$\mathrm{MAE} = \frac{1}{M}\sum_{i=1}^{M} |\, b_i - \mu_i \,|,$$

where M is the number of the matched bursts

b_i base value for burst i

$\mu_i = \frac{1}{N-1}\sum_{j \neq base} t_{j,i}$

Relative Difference

Relative deviation for non-zero base values

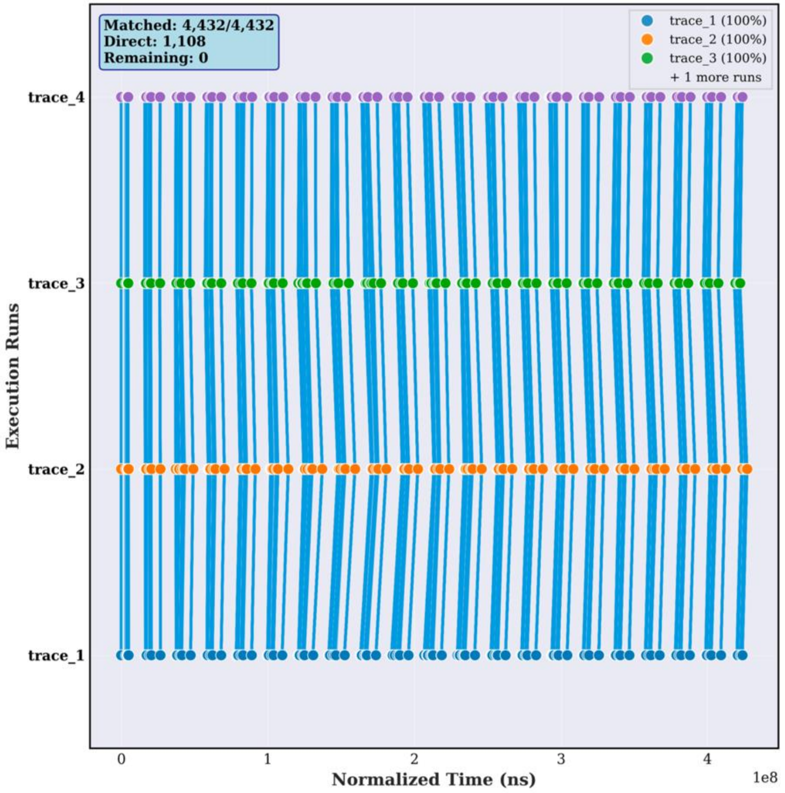
`< 30% Acceptance Rate`

$$\mathrm{RelDiff} = \frac{1}{|\mathcal{I}|}\sum_{i \in \mathcal{I}} \frac{|\, b_i - \mu_i \,|}{b_i}, \qquad \mathcal{I} = \{\, i : \, b_i > 0 \,\}.$$

Experimentation Validation

Applications with Identical Communication Behaviour

Validation approach using Lulesh with identical counter sets extracted from 4 different executions.



Lulesh application for TaskID 9

LULESH *INS_MIX* VALIDATION

Counter	Correlation	MAE	Rel Diff	< 30% Diff
PAPI_TOT_INS	1.000	39.789	0.011	100.0%
PAPI_SR_INS	1.000	6.914	0.014	100.0%
PAPI_LD_INS	1.000	12.193	0.011	100.0%
PAPI_BR_INS	1.000	4.690	0.007	100.0%
PAPI_TOT_CYC	1.000	349.238	0.081	96.7%
PAPI_L1_DCM	1.000	12.651	0.204	76.3%
PAPI_L2_DCM	1.000	25.171	0.390	61.7%
PAPI_L3_TCM	0.973	11.387	0.499	51.6%

For the OPS SET and OPS CYC counter sets, detailed validation tables are not presented since these applications achieved **perfect deterministic matching** with no duplicate columns generated during trace fusion.

Experimentation Validation

Processes with Estructural Variations Between Executions

SOD2D *INS_MIX* VALIDATION RESULTS

Counter	Correlation	MAE	Rel Diff	< 30% Diff
PAPI_L2_DCM	1.000	2.729	0.512	50.8%
PAPI_L1_DCM	0.999	7.165	0.214	76.1%
PAPI_L3_TCM	0.999	2.000	0.488	51.3%
PAPI_SR_INS	0.963	7.387	0.010	99.9%
PAPI_LD_INS	0.876	13.693	0.009	99.7%
PAPI_BR_INS	0.875	6.258	0.005	99.7%
PAPI_TOT_INS	0.876	20.070	0.003	99.7%

Highly Irregular Traces

SEISSOL *INS_MIX* VALIDATION

Counter	Correlation	MAE	Rel Diff	< 30% Diff
PAPI_SR_INS	0.994	4.942	0.015	97.2%
PAPI_LD_INS	0.993	11.373	0.015	97.3%
PAPI_TOT_INS	0.993	41.992	0.019	97.3%
PAPI_L1_DCM	0.991	4.738	0.669	38.5%
PAPI_BR_INS	0.989	4.725	0.008	97.4%
PAPI_L2_DCM	0.980	0.885	0.092	87.2%
PAPI_L3_TCM	0.976	1.129	0.174	75.7%

SEISSOL *OPS_SET* VALIDATION RESULTS

Counter	Correlation	MAE	Rel Diff	< 30% Diff
PAPI_VEC_SP	1.000	0.000	0.000	100.0%
PAPI_SP_OPS	1.000	0.000	0.000	100.0%
PAPI_VEC_DP	0.993	0.000	0.000	99.8%
PAPI_VEC_INS	0.993	0.000	0.000	99.8%
PAPI_DP_OPS	0.993	0.000	0.000	99.4%
PAPI_FP_OPS	0.993	0.000	0.000	99.4%
PAPI_FP_INS	0.992	0.000	0.000	99.4%

Conclusions and Future Work

Conclusions

Research Objectives

1. Overcome **the n-counter ceiling** in HPC performance modeling.
2. Maintain **burst-level fidelity** required for detailed analysis.
3. Create synthetic traces compatible with **existing tools** (Paraver/Extrae).
4. Validate methodology across **diverse HPC applications.**

Conclusions

- **Excellent Performance**
  - Instruction counters: >97% acceptable correspondence
  - Floating-point operations: Perfect precision even in non-deterministic scenarios
- **Variable Performance**
  - Cache hierarchy: 38.5-87.2% acceptance rates
  - Treat cache data as supplementary evidence requiring interpretation

Future Work

1. **Training performance prediction models** on expanded feature spaces and expand our prior work on performance analysis of HPC workloads.

2. **Production deployment** requires evaluation across broader application domains and HPC architectures to confirm generalization.

Thank you!