

FAIR Publication of HPC Performance Data

*** Alex Wiens, † Shakoor Pooseh, † Matthias Lieber**

* Paderborn Center for Parallel Computing, Paderborn University

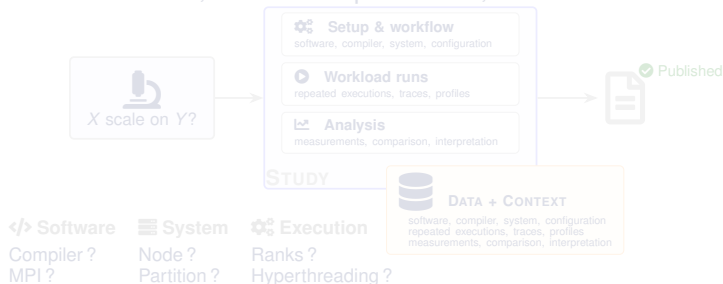
† CIDS, TU Dresden

International Parallel Tools Workshop, Dresden, 02 Oct. 2026

- ❑ Why performance data usually is **not** reusable
- ❑ The three-part plan: collect, standardize, publish
- ❑ **Metaper**, automatic metadata capture on the cluster
- ❑ **Metadata schema**, everything in sync
- ❑ **Perfortal**, HPC performance data exchange portal
- ❑ Vision: Research Data Management Containers (RDMC)
- ❑ Open questions and how you can help

The Problem: data without context

- ❑ HPC centers produce **many traces and profiles per day**
- ❑ Almost none of this data is shared outside the group that made it
- ❑ After six months, even with a publication, data is hard to interpret



- ❑ Data survives but the context is often missing or incomplete

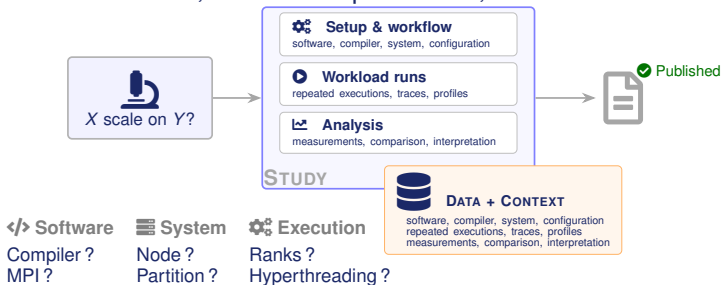
❑ HPC centers produce **many traces and profiles per day**

- 35s from a couple of minutes, 32 GiB



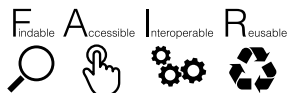
The Problem: data without context

- ❑ HPC centers produce **many traces and profiles per day**
- ❑ Almost none of this data is shared outside the group that made it
- ❑ After six months, even with a publication, data is hard to interpret



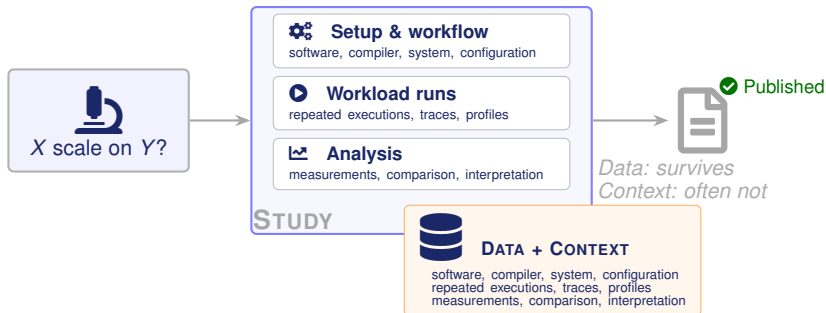
- ❑ Data survives but the context is often missing or incomplete

Why This Matters: FAIR for performance data

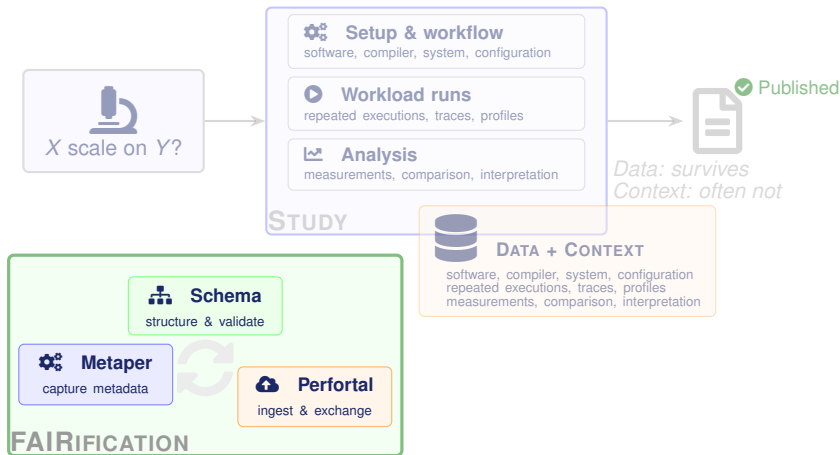


- ❑ For performance data, **Interoperable** and **Reusable** break first
 - ❑ A trace is technically *accessible* if it's emailed to you
 - ❑ It's not *reusable* without the metadata to reconstruct how it was made
- ❑ Our approach
 - ❑ Treat *metadata capture* as the main lever for *FAIRness*
 - ❑ **The paper preserves the scientific result**
 - ❑ **We make the performance data reusable**

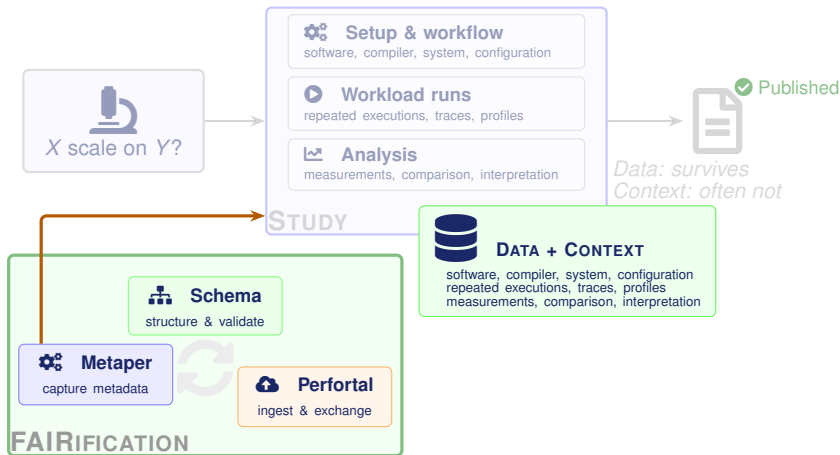
FAIR performance data



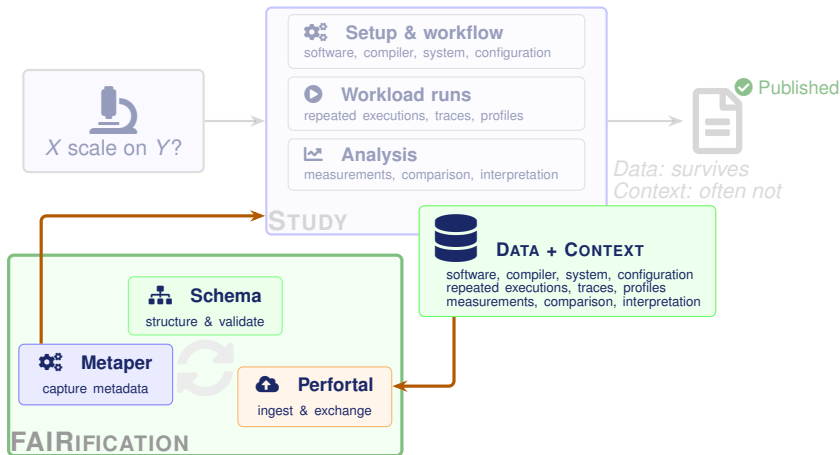
FAIR performance data



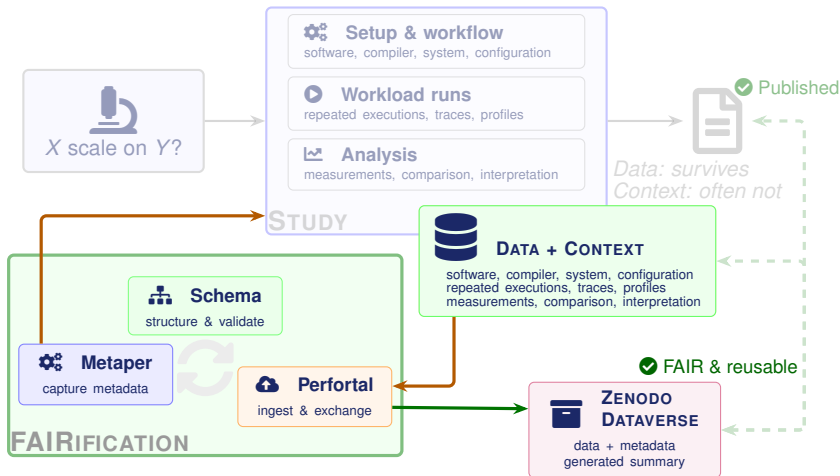
FAIR performance data



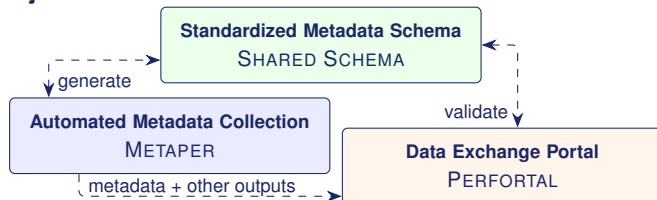
FAIR performance data



FAIR performance data



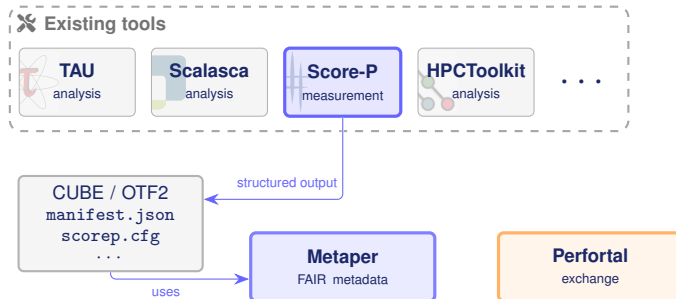
Three connected objectives:



Design philosophy: enrichment is optional

- ❑ Run `metaper` with **zero configuration** and get `metadata.json`
- ❑ Upload to Perfortal with **that file**
- ❑ Add dataset info, tool-specific summaries, eventually an RDMC
- ❑ Each step is **opt-in**
- ❑ Mirrors the FAIR principles of a **maturity model**, not a pass/fail test

Building on an existing HPC performance ecosystem

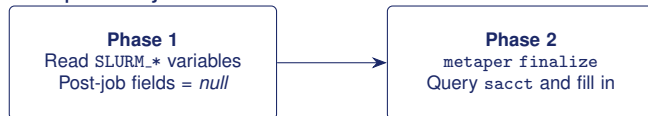


Metaper: design goals

- ❑ Minimal setup required
 - ❑ One `metaper` invocation
 - ❑ No schema knowledge up front
- ❑ Two-phase capture
 - ❑ Some data exists during the job
 - ❑ Some only after the scheduler terminates the job
- ❑ Extensible via plugins
 - ❑ Tool-specific postprocessing (Score-P today) without touching the core
- ❑ Anonymization as a first-class option

Metaper: architecture

- ❑ Python package
 - ❑ Built on top of *MachineState*
 - ❑ Organized into `core/`, `job/`, `hardware/`, ...
- ❑ Hardware info collected **per node** and compressed (raw kept)
- ❑ Errors as values
 - ❑ Fields that cannot be collected are written to `metadata_error.json`
- ❑ Two-phase "job" info collector



- ❑ Optional human-supplied info about the dataset in CWD, `dataset_metadata.json`

```
"name": "npb-mz-c-benchmark"
"workloadFamily": "computational_fluid_dynamics"
"libraries": "MPI, OpenMP, Score-P"
```

Metaper: pipeline overview

```
BUILD_RUN.SBATCH
```

```
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4

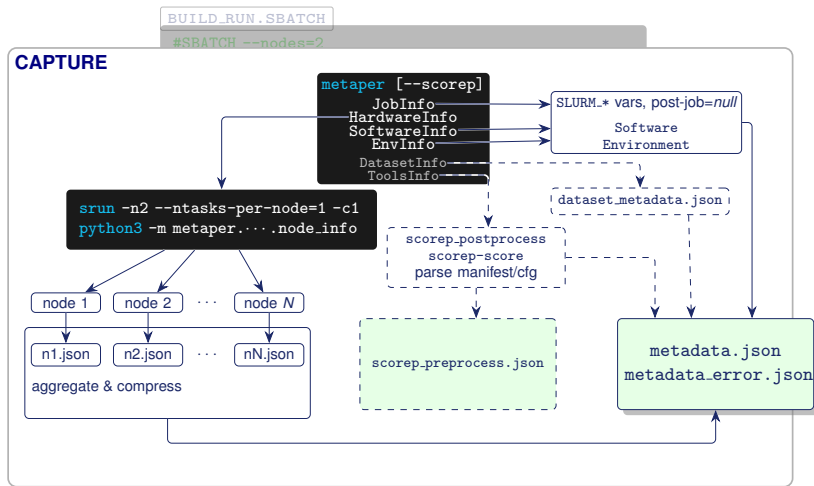
# Module load & venv
ml release/2026 GCC/14.3.0 OpenMPI/5.0.8 Score-P/9.4
source /path/to/venv/bin/activate

# Build
make PREP=scorep

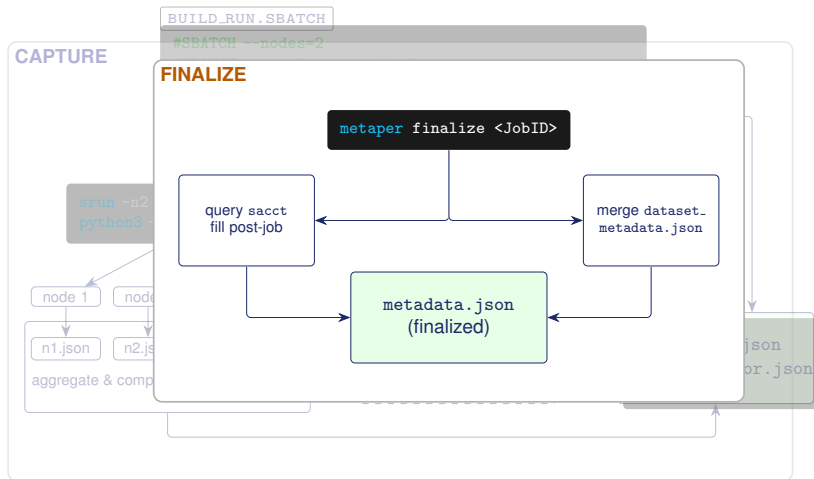
# Workload
srun -n 8 -c 1 heat-blocking.scorep 4096 4096 128

# Metadata
metaper --scorep
```

Metaper: pipeline overview



Metaper: pipeline overview



Metaper: what you get

```
demo_heat/
├─ metaper_<JobID>
│   ├── config.json ..... parsed flags (debug)
│   ├── metadata.json ..... main output
│   ├── metadata_error.json ..... per-field errors, if any (debug)
│   ├── scorep_preprocess.json ..... tool-specific main output
│   ├── hardware_specs ..... per-node hardware (debug)
│   │   ├── 24
│   │   │   └─ node_i8019.json
│   │   ├── 87
│   │   │   └─ node_i8018.json
│   └─ summary_scorep ..... tool-specific outputs (debug)
│       ├── scorep-score.txt
│       ├── scorep_score.json
│       ├── scorep_metadata.json
│       └─ summary.md
```

data/metada

```

"topology": {
  "mpiRanks": 8,
  "threadsPerRank": 1,
  "ranksPerNode": 4,
  "gpusPerNode": 2,
  "gpusPerTask": null
},
"timing": {
  "startTime": "2026-07-2",
  "submitTime": "2026-07-",
  "eligibleTime": "2026-0",
  "endTime": "2026-07-21T",
  "wallTimeSeconds": 33,
  "cpuTimeSeconds": 528,
  "queueWaitTimeSeconds":
},
"usage": {
  "maxResidentMemoryBytes",
  "maxVirtualMemoryBytes",
  "maxDiskReadBytes": 280,
  "maxDiskWriteBytes": 43,
  "maxPageFaultCount": 28
},
"termination": {
  "state": "COMPLETED",
  "exitCode": 0,
  "derivedExitCode": 0,
  "signal": null
}
},
"hardware": {
  "cpu": [
    {
      "id": "cpu-6e0185",
      "vendor": "Authenti",
      "modelName": "AMD E

```

```
    "family": 23,  
    "model": 49,  
    "stepping": 0,  
    "baseFrequencies": [  
        "3.20 GHz"  
    ],  
    "sockets": 2,  
    "coresPerSocket": 24,  
    "totalCores": 48,  
    "smtActive": true,  
    "smtWidth": 2,  
    "totalThreads": 96,  
    "numaNodes": 4,  
    "cache": {  
        "L1d": "32 KiB",  
        "L1i": "32 KiB",  
        "L2": "512 KiB",  
        "L3": "16 MiB"  
    }  
},  
    },  
    "memory": {  
        "id": "mem-f9102b",  
        "memTotal": "984 GiB",  
        "swapTotal": "0 B",  
        "hugepageSize": "2 MiB",  
        "hugepagesTotal": 0,  
        "hugepagesMemory": "0 B"  
    }  
},  
    "gpu": {  
        "id": "gpu-a92f26",  
        "brand": "nvidia",  
        "productName": "NVIDIA A100-SXM4-40GB"
```

Metaper: tool-specific postprocessing

- ❑ First plugin: `scorep_postprocess`
 - ❑ Runs `scorep-score`
 - ❑ Parses `manifest.json` / `scorep.cfg`
 - ❑ MPI / USER / COM region breakdowns
 - ❑ Top N most time-consuming functions
 - ❑ Percentage of runtime breakdowns
- ❑ Invoked as `--scorep topn=[10], [hash]` on the command line
- ❑ A lightweight summary of the full trace
- ❑ Plugins are self-contained, future tools without touching this one

Metaper: tool-specific JSON output

```
{
  "$schema": "https://gitlab.gwdg.de/mpi/mpi/-/tree/main/scorep/schemas/metadata",
  "schemaVersion": 1,
  "generator": {
    "name": "Metaper",
    "version": "0.1.0",
    "generatedAt": "2026-07-21T11:50:00Z"
  },
  "tool": "scorep",
  "overview": {
    "totalTimeSeconds": 32.15,
    "totalVisits": 7562
  },
  "performance": {
    "timeDistribution": {
      "mpi": 85.72,
      "com": 0.03,
      "usr": 14.25,
      "scorep": 0.0
    },
    "topFunctions": [
      {
        "type": "MPI",
        "region": "MPI_Init",
        "visits": 8,
        "time": 25.26,
        "timePercent": 78.57
      },
      {
        "type": "USR",
        "region": "heatTimestepInner",
        "visits": 1024,
        "time": 4.47,
        "timePercent": 13.9
      }
    ]
  }
}
```

```
    {
      "type": "MPI",
      "region": "MPI_Waitall",
      "visits": 1024,
      "time": 0.97,
      "timePercent": 3.02
    },
    {
      "type": "MPI",
      "region": "MPI_Barrier",
      "visits": 16,
      "time": 0.51,
      "timePercent": 1.59
    },
    {
      "type": "MPI",
      "region": "MPI_Send",
      "visits": 42,
      "time": 0.4,
      "timePercent": 1.24
    },
    {
      "type": "MPI",
      "region": "MPI_Recv",
      "visits": 42,
      "time": 0.29,
      "timePercent": 0.9
    },
    {
      "type": "USR",
      "region": "heatTimestepBound",
      "visits": 1024,
      "time": 0.06,
      "timePercent": 0.19
    },
    {
      "type": "MPI",

```

```
      "region": "MPI_Startall",
      "visits": 1024,
      "time": 0.05,
      "timePercent": 0.16
    },
    {
      "type": "MPI",
      "region": "MPI_Finalize",
      "visits": 8,
      "time": 0.05,
      "timePercent": 0.16
    },
    {
      "type": "USR",
      "region": "heatTotalEnergy",
      "visits": 2,
      "time": 0.04,
      "timePercent": 0.12
    }
  ]
}
```

s/scorep/s

Metaper: installation

- ❑ Once, on the login node

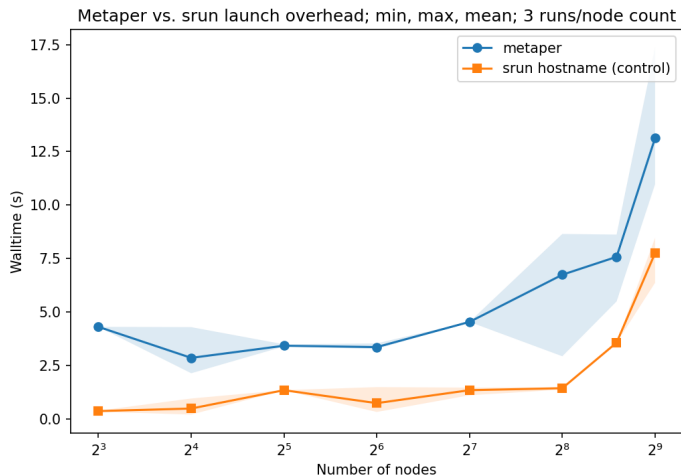
```
ml release/2026 GCC/14.3.0 Python/3.13.5
python3 -m venv path/to/v_env
source path/to/v_env/bin/activate
pip install path/to/metaper
```

- ❑ Inside every batch script, same modules as install time

```
ml release/2026 GCC/14.3.0 Python/3.13.5
source path/to/v_env/bin/activate
```

- ❑ A little extra work to ensure the functionality
 - ❑ Future: admin installation via EasyBuild or so

Metaper: scalability



Metaper: status and availability



GPL-3.0

`gitlab.gwdg.de/.../metaper`

✔ Working today

test suite · TESTING.md

documentation

2D MPI heat-equation solver

DEMO

NAS BT-MZ (MPI+OpenMP)

⌚ In development

more tools

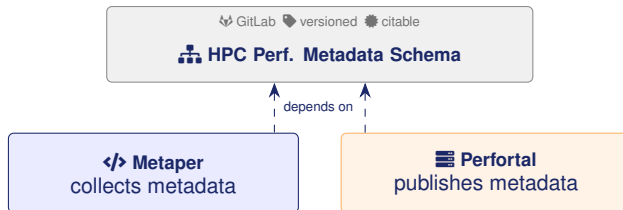
tools-specific reduced formats

HPC-native installation

What is a schema?

- Formally defines the *structure, types, and constraints* of a data format – a contract between producers and consumers of the data
- We use **JSON Schema** (Draft 2020-12):
machine-readable $\Rightarrow$ validation, tooling, self-documenting etc.
- Why: consistent metadata across schedulers/sites, automated validation & interoperability, a versioned and citable artifact (FAIR)

```
"identity": {
  "title": "Job Identity",
  "type": "object",
  "required": ["jobId", "scheduler"],
  "properties": {
    "jobId": {
      "type": "string",
      "description": "Unique identifier assigned to the job by
                    the scheduler, e.g. ...",
    },
    "clusterName": {
      "type": "string",
      "examples": ["barnard", "otus"]
    }
  }
}
```



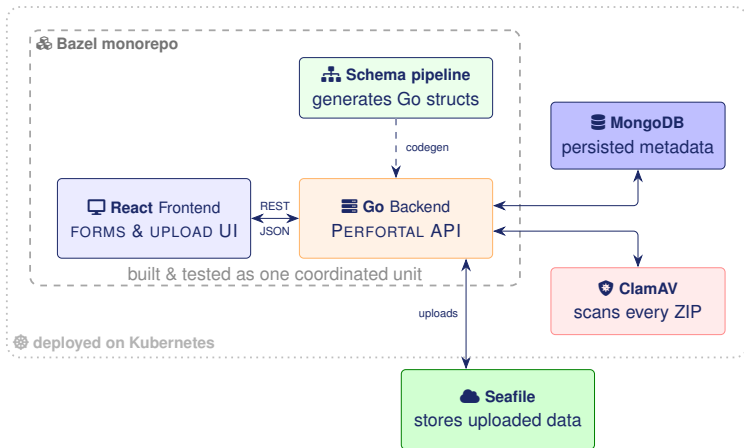
- ❑ HPC performance metadata schema

`gitlab.gwdg.de/nfdixcs/i3-perf.../hpc-performance-metadata`

- ❑ A citable, independently versioned artifact

- ❑ Both Metaper and Perfortal consume it as an external dependency

Perfortal: architecture [based on hpcioanalysis.zdv.uni-mainz.de]



Perfortal: upload workflow

✉ 1. Email verification

minimal identity for attribution & follow-up
accepts email from German institutes

● Contact ⓘ

● Tools & Imports

● Dataset

● Workload

● Cluster

● HW/SW

● Submit

During the pilot phase, Perfortal accepts email addresses from most German universities and publicly funded research institutions. If your access is denied, please **contact us**.

Perfortal: upload workflow

1. Email verification

2. Import from Metaper

metadata.json

scorep_preprocess.json

☒ Contact

☒ Tools & Imports

☐ Dataset

☐ Workload

☐ Cluster

☐ HW/SW

☐ Score-P

☐ Submit

Select every tool used for this experiment. If you have Metaper output, upload it here to pre-fill the matching steps -- everything stays editable afterward, and nothing here is required to continue.

Metadata file



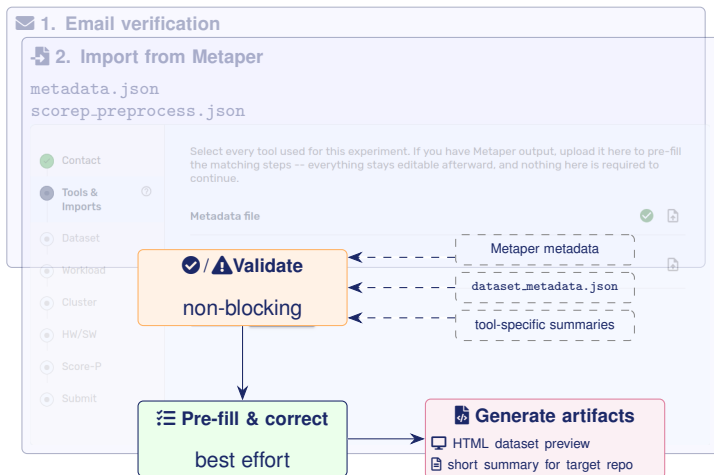
☒ Score-P



BACK

CONTINUE

Perfortal: upload workflow



Perfortal: upload workflow

1. Email verification

2. Import from Metaper

3. Review & correct metadata

mandatory + optional form fields

✓ Contact

✓ Tools & Imports

✓ Dataset

✓ Workload

✓ Cluster

✓ HW/SW

● Score-P

?

○ Submit

Score-P Version

Score-P 9.2

MPI Time (%)

86

User Time (%)

14

☒ OTF2 Tracing

BACK

CONTINUE

20 / 27

Perforal: upload workflow

1. Email verification

2. Import from Metaper

3. Review & correct metadata

4. Submit

source, scripts, traces, summaries

✓ Contact

✓ Tools & Imports

✓ Dataset

✓ Workload

✓ Cluster

✓ HW/SW

✓ Score-P

● Submit ⓘ

Upload Size: Archives should not exceed **500 MB**, and the extracted content should stay under **700 MB**.

File Types: ZIP and TAR archives (.zip, .tar, .tar.gz, .tgz) are accepted. Compiled binaries, test scripts, or unrecognized shared libraries may trigger false-positive malware alarms. Please try removing executables or non-essential logs before re-uploading.

SELECT ARCHIVE (ZIP OR TAR < 500 MB, EXTRACTED < 700 MB)

☐ I agree to the **terms & conditions** *

↑ SUBMIT

BACK

Perforal: upload workflow

1. Email verification

2. Import from Metaper

3. Review & correct metadata

4. Submit

source, scripts, traces, summaries

Contact

Tools & Imports

Dataset

Workload

Cluster

HW/SW

Score-P

Submit

Upload Size: Archives should not exceed **500 MB**, and the extracted content should stay under **700 MB**.
File Types: ZIP and TAR archives (.zip, .tar, .tar.gz, .tgz) are accepted. Compiled binaries, test scripts, or unrecognized shared libraries may trigger false-positive malware alarms. Please try removing executables or non-essential logs before re-uploading.

SELECT ARCHIVE (ZIP OR TAR < 500 MB, EXTRACTED < 700 MB)

☐ I agree to the **terms & conditions** *

 **SUBMIT**

BACK

 700 MB each

 5T cloud storage now

 Dataset registered in Perforal

Perforal: upload workflow

1. Email verification

2. Import from Metaper

3. Review & correct metadata

4. Submit

source, scripts, traces, summaries

Contact

Tools & Imports

Dataset

Workload

Cluster

HW/SW

Score-P

Submit

Upload Size: Archives should not exceed **500 MB**, and the extracted content should stay under **700 MB**.

File Types: ZIP and TAR archives (.zip, .tar, .tar.gz, .tgz) are accepted. Compiled binaries, test scripts, or unrecognized shared libraries may trigger false-positive malware alarms. Please try removing executables or non-essential logs before re-uploading.

SELECT ARCHIVE (ZIP OR TAR < 500 MB, EXTRACTED < 700 MB)

☐ I agree to the **terms & conditions** *

SUBMIT

BACK

700 MB each

5T cloud storage now

Dataset registered in Perforal

Available by an admin approval

Perfortal: dataset preview



Datasets**Upload**

Filters

Columns Full screen

	Dataset Name	Workload Family	System Organization	Nodes • Cores GPUs	CPU Accelerator	Comm • Compute %	Walltime
DETAILS	heat-conduction-mpi-demo	MPI Demo	Alpha TU Dresden	2N • 8C 4G	AMD EPYC 7352 24-Core Processor NVIDIA A100-SXM4-40GB	86  14	33s
	Uploaded 24 days ago	Size 266.3 KiB	Submitter Shakoor Pooseh	Application Heat Conduction Miniapp	Cluster / partition Alpha / Alpha	Libraries MPI, Score-P	

Rows per page 10 1-1 of 1 < >

About

The HPC Performance Data Exchange Portal is part of the NFDIXCS initiative, aiming to support the research community with open, structured access to performance data and metadata from high-performance computing environments.

Built upon the software and design of the HPC IO Analysis web server (© 2024)

Links

[NFDIXCS](#)[HPC IO Analysis](#)[TU Dresden](#)

[Impressum](#)[Terms of Use](#)[Contact Us](#)

Perforal: dataset preview

 **heat-conduction-mpi-demo** Job #2301782

[DOWNLOAD](#) [REPORT](#) [SHARE](#) [CLOSE](#)

Heat Conduction Miniapp • alpha • pooseh • 2026-07-21T11:57:22.588622+00:00

[MPI](#) [Score-P](#) [Score-P 9.2](#) [2 nodes • 8x CPU](#) [4x GPU](#) [alpha](#)

OVERVIEW

HARDWARE

PERFORMANCE

ARTIFACTS

RUNTIME

32.1

seconds

MPI OVERHEAD

85.7%

of runtime

TOTAL VISITS

7.562

call-path entries

TRACE SIZE

457.09 KB

JOB RESOURCES

Node count	2
CPU count	8
GPU count	4
Memory per node	64 GiB
Queue	alpha

SOFTWARE STACK

OS	AlmaLinux 9.8 (Olive Jaguar)
Score-P	Score-P 9.2
Timer	tsc
Tracing	enabled (OTF2)
Profiling	enabled (Cube4)

Perfortal: dataset preview

The screenshot displays the 'heat-conduction-mpi-demo' dataset preview in the Perfortal portal. The interface includes a top navigation bar with 'Datasets' and 'Upload' tabs. The main content area shows the dataset title, job ID, and a list of tags (MPI, Score-P, Score-P 9.2, 2 nodes · 8x CPU, 4x GPU, alpha). Below this are tabs for Overview, Hardware, Performance, and Artifacts. The Hardware tab is active, showing details for CPU, GPU, Memory, and Network.

heat-conduction-mpi-demo Job #2301782

Heat Conduction Miniapp · alpha · pooseh · 2026-07-21T11:57:22.588622+00:00

MPI **Score-P** **Score-P 9.2** **2 nodes · 8x CPU** **4x GPU** **alpha**

OVERVIEW **HARDWARE** PERFORMANCE ARTIFACTS

CPU

Model	AMD EPYC 7352 24-Core Processor
Cores	48 (2 x 24)
L3 cache	16 MiB
NUMA nodes	4
SMT	enabled
Vendor	AuthenticAMD

GPU

Model	NVIDIA A100-SXM4-40GB
VRAM	40 GiB
SM clock	1410 MHz
Compute capability	8.0
PCIe	Gen4 x16
ECC	enabled
Power limit	400 W
Driver	580.159.04

MEMORY

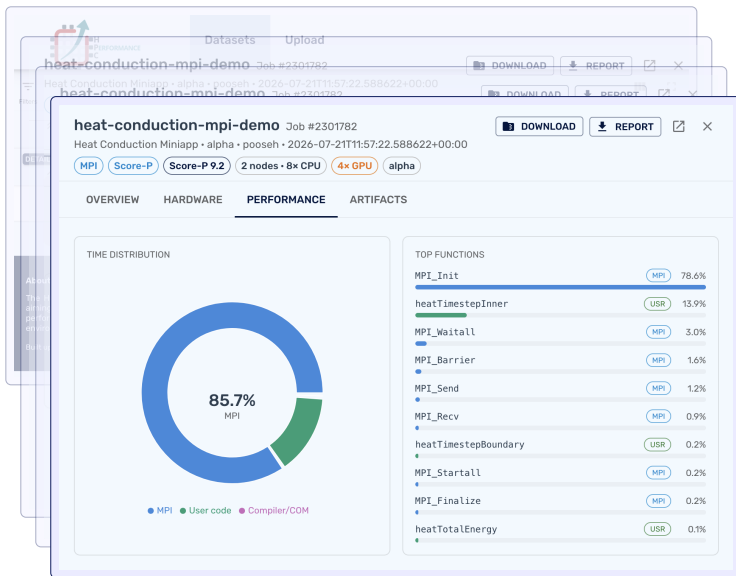
Total RAM	984 GiB
Hugepages	none
Swap	none

NETWORK

InfiniBand	200 Gb/sec (4X HDR)
IB firmware	20.43.8004
Fast ethernet	200 Gb/s
Ethernet driver	mlx5_core

Publishing HPC Perf. Experiments

Perforal: dataset preview



Publishing HPC Perf. Experiments

Perforal: dataset preview

The screenshot displays the 'heat-conduction-mpi-demo' dataset preview in the Perforal application. The interface includes a sidebar with navigation options like 'About', 'The dataset', 'Performance', and 'Build'. The main content area shows the dataset title, job ID, and a list of artifacts. The 'ARTIFACTS' tab is active, showing a table of files and their sizes. Below the table, there are 'SCORE-P CONFIGURATION HIGHLIGHTS'.

heat-conduction-mpi-demo Job #2301782

Heat Conduction Miniapp • alpha • pooseh • 2026-07-21T11:57:22.588622+00:00

[MPI](#) [Score-P](#) [Score-P 9.2](#) [2 nodes • 8x CPU](#) [4x GPU](#) [alpha](#)

OVERVIEW **HARDWARE** **PERFORMANCE** **ARTIFACTS**

../scorep-20260721_1357_1399431550472949
21 files • 679.59 KB

MANIFEST.md	Manifest	976.00 B
profile.cubex	Cube4 profile	205.00 KB
scorep.cfg	Score-P config	1.65 KB
traces/	OTF2 event streams	16 files 457.09 KB
traces.def	Global defs	14.61 KB
traces.otf2	OTF2 anchor	283.00 B

SCORE-P CONFIGURATION HIGHLIGHTS

SCOREP_ENABLE_PROFILING	true
SCOREP_ENABLE_TRACING	true
SCOREP_TOTAL_MEMORY	32M
SCOREP_TIMER	tsc
SCOREP_PROFILING_FORMAT	cube4
SCOREP_PROFILING_ENABLE_CLUSTERING	true

Perfortal: status and availability

 `perfortal.zih.tu-dresden.de`

✔ Working today

✔ core upload & pre-fill

✔ metadata model

✔ non-blocking schema validation

roadmap



⌚ In development

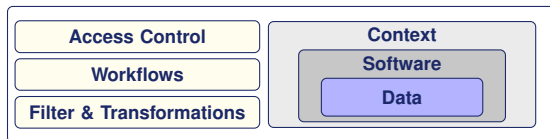
+ more tools

🏗 RDMC integration

📄 Dataverse / Zenodo export

What is NFDI x CS?

- ❑ National **F**orschungs**d**aten**i**nfrastruktur for and with **C**omputer **S**cience, one consortium of Germany's NFDI initiative
- ❑ Goal: interoperable infrastructure for FAIR data and software exchange across CS
- ❑ Introduces **RDMC**



Why connect to the RDMC

- ❑ Metaper + schema + Perfortal form:
 - ❑ a useful self-contained pipeline
 - ❑ an HPC-specific FAIR exchange
- ❑ NFDIxCS's RDMC forms a **common** unit of exchange across CS
- ❑ An RDMC is a bundle of interconnected research artifacts plus the execution context
 - ❑ Data Artifacts
 - ❑ Software Artifacts
 - ❑ Execution Environment (REE)
 - ❑ Metadata Manifest
 - ❑ External Links (DOI/Repository)
- ❑ A dataset that is FAIR within Perfortal is not automatically interoperable unless it becomes an RDMC

- ❑ Tool coverage is Score-P only
 - ❑ add support for other tools e.g. TAU, HPCToolkit, PIKA?
- ❑ Anonymization could be improved
 - ❑ currently: user names, IP addresses in the metadata
 - ❑ Needs to address re-identification from data
- ❑ Perfportal is a single TU Dresden hosted instance
 - ❑ Federation across centers
- ❑ RDMC generation



Metaper

- ❑ Automatic, two-phase, plugin-extensible metadata capture for HPC jobs
`gitlab.gwdg.de/nfdixcs/i3-performance-management`
- ❑ Try out the demo contained in the repository on your cluster

Shared schema

- ❑ One source of truth
`gitlab.gwdg.de/nfdixcs/i3-performance-management`

Perfortal

- ❑ Adds metadata, validates, enriches, and publishes (Dataverse/Zenodo)
`perfortal.zih.tu-dresden.de`

NFDIxCS

- ❑ `https://nfdixcs.org`
Funding information: DFG NFDI 52/1, Project No. 501930651



2-minute survey:

- ☐ Tool coverage priorities
- ☐ Would you use Perfortal?

tud.link/vvhzym