ADMIRE User Day

Extra-P With I/O Modeling Capabilities

Ahmad Tarraf
Technical University Darmstadt

December 12th 2023

Barcelona Supercomputing Center

TECHNISCHE UNIVERSITÄT DARMSTADT
Parallel Programming

Grant Agreement number: 956748 — ADMIRE — H2020-JTI-EuroHPC-2019-1

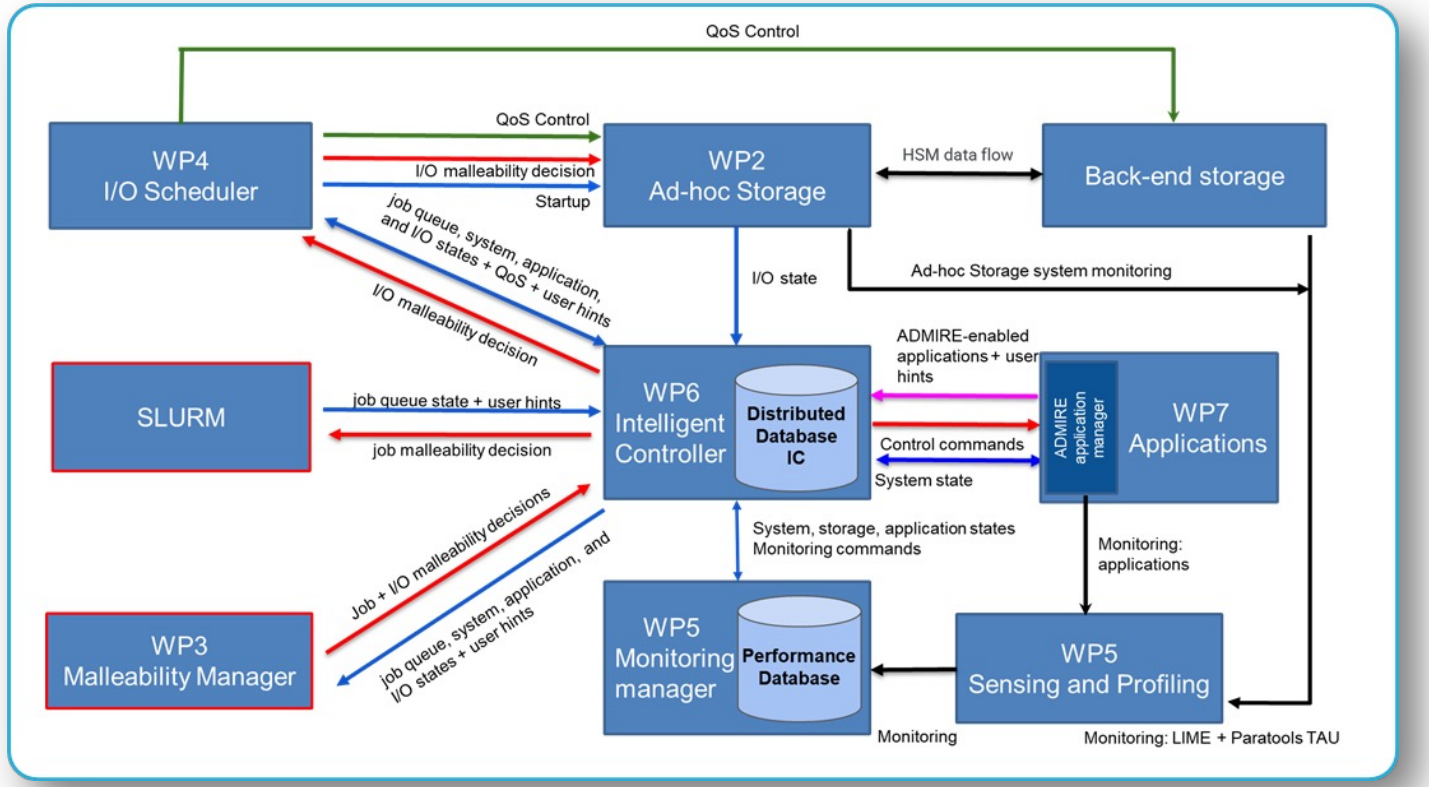
EuroHPC Joint Undertaking

Performance Models in ADMIRE

ADMIRE global architecture:

- Intelligent controller acts as the main component forwarding and processing data
- Key aspects examined in the ADMIRE project is **job malleability**

- Malleability manager needs:
  - Current state of the system
  - Jobs in the queue
  - User hints
  - **Scaling performance of an application**



Objective: balance both compute and I/O resources
→Performance models that cover the scaling behavior applications in both regards

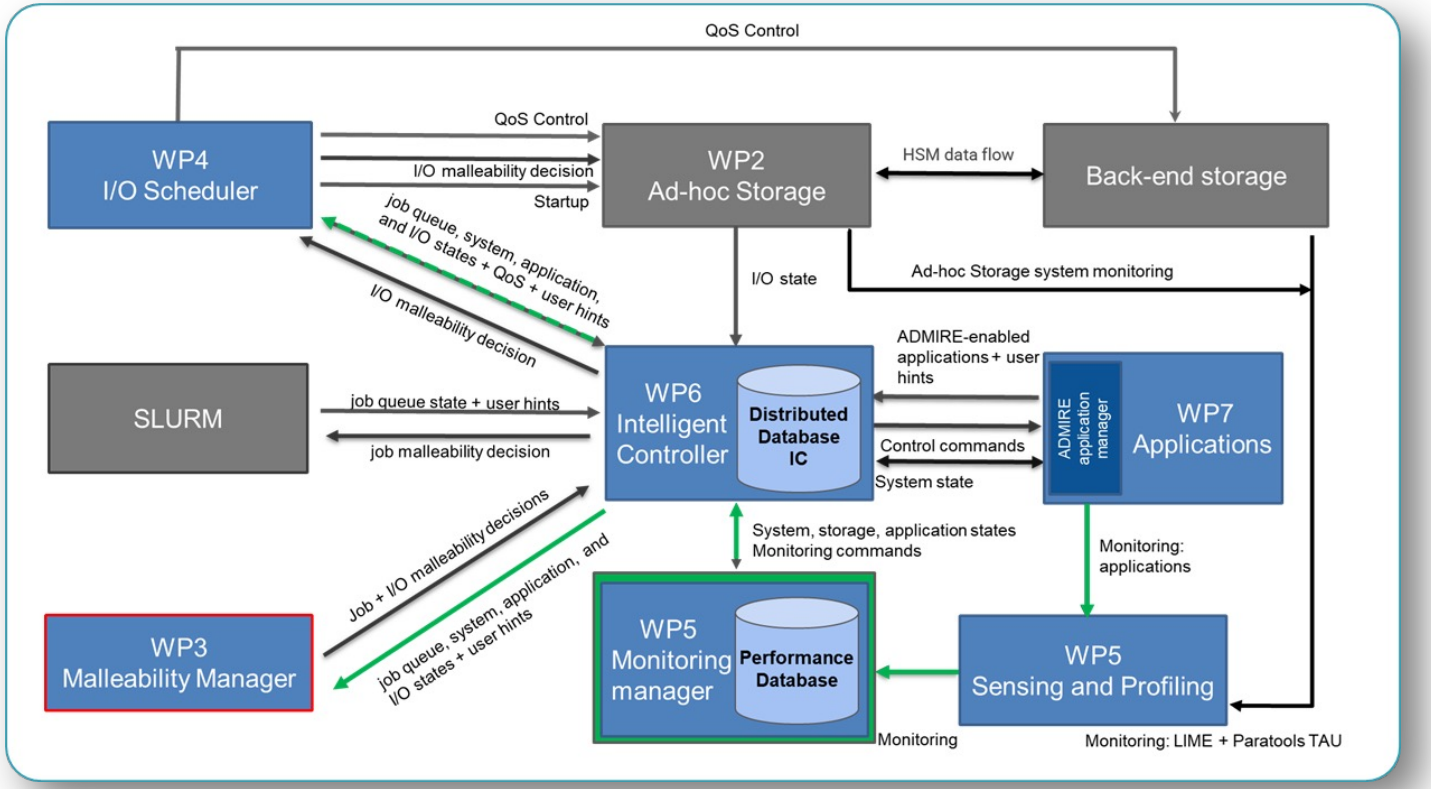
Performance Models in ADMIRE (cont')

Generate performance models with Extra-P

- Components that directly influence or interact with the performance models from Extra-P are colored (in blue)
- Data flow of the models is colored in green

So far, Extra-P focuses on modeling the scaling behavior of computational and communicational parts of an applications

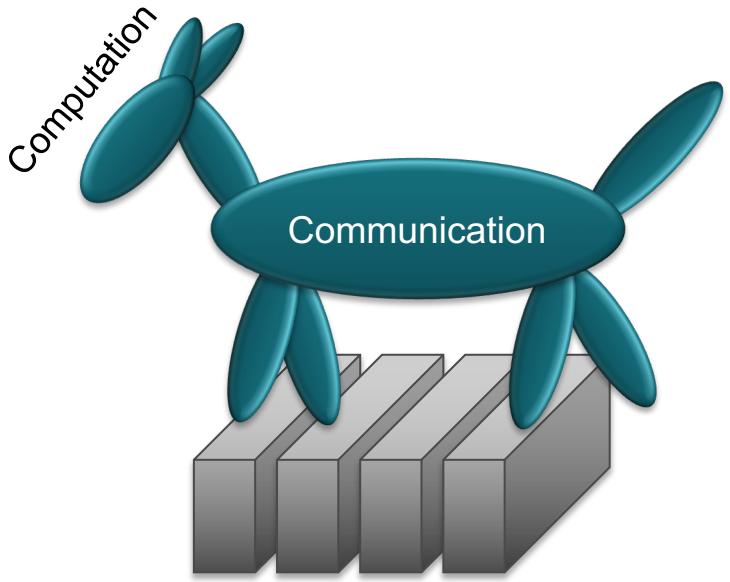
Extend Extra-P to model I/O!

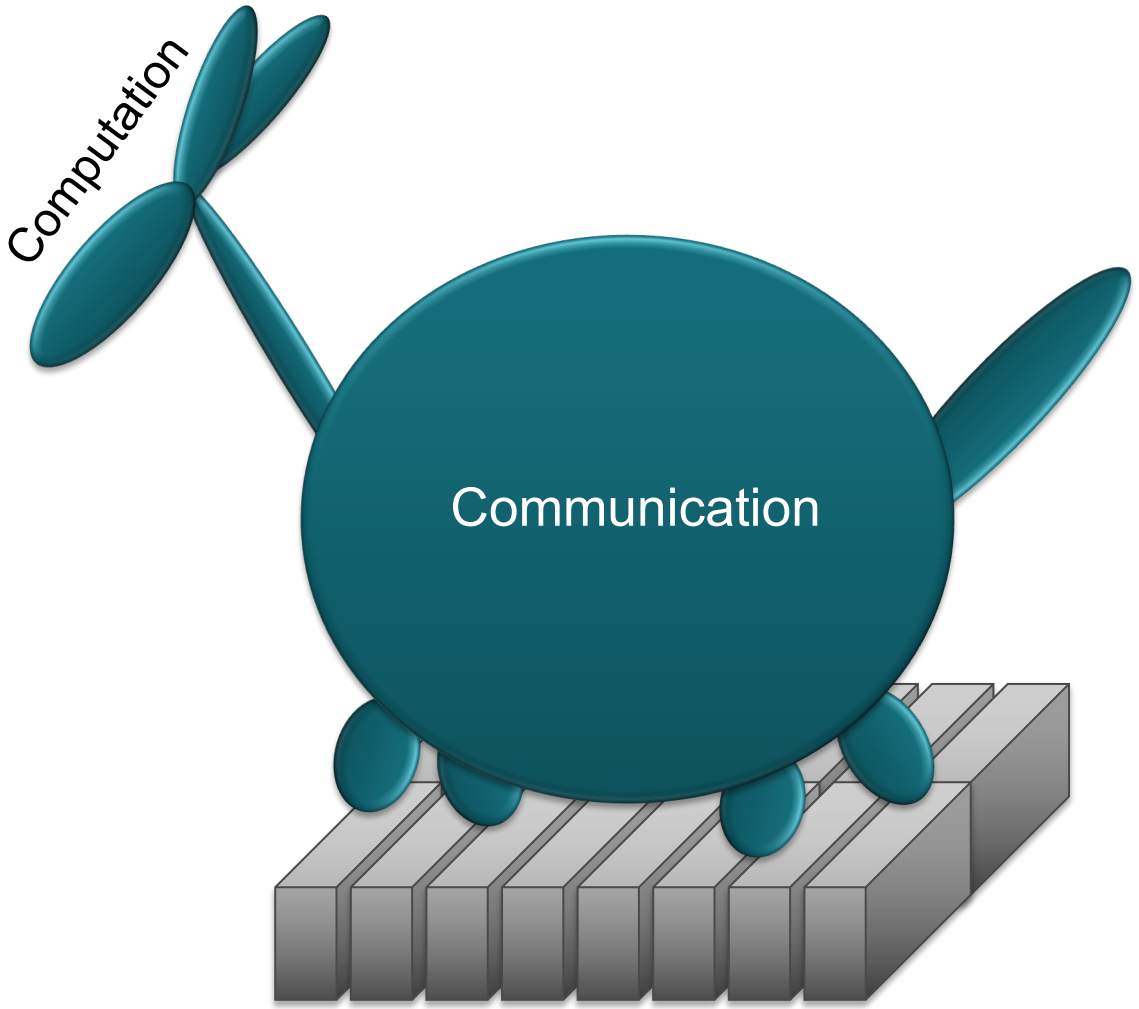


But before, let's have a quick look at Extra-P and performance models in general …

Scaling your code can harbor performance surprises*…

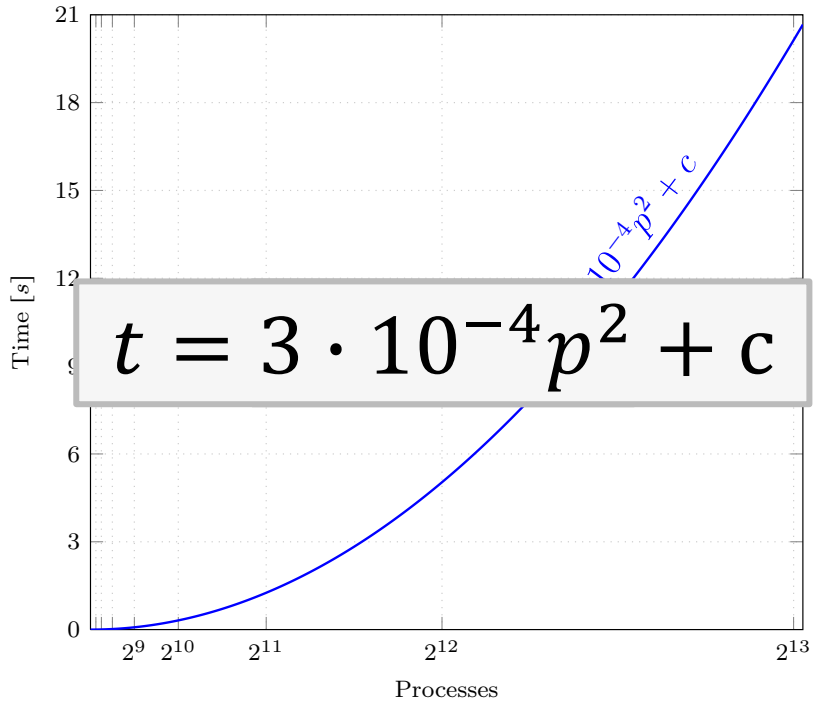
*Goldsmith et al., 2007





Performance Models

Performance Model: Formula that expresses a relevant performance metric as a function of one or more execution parameters



Analytical (i.e., manual) creation
→ challenging for entire programs

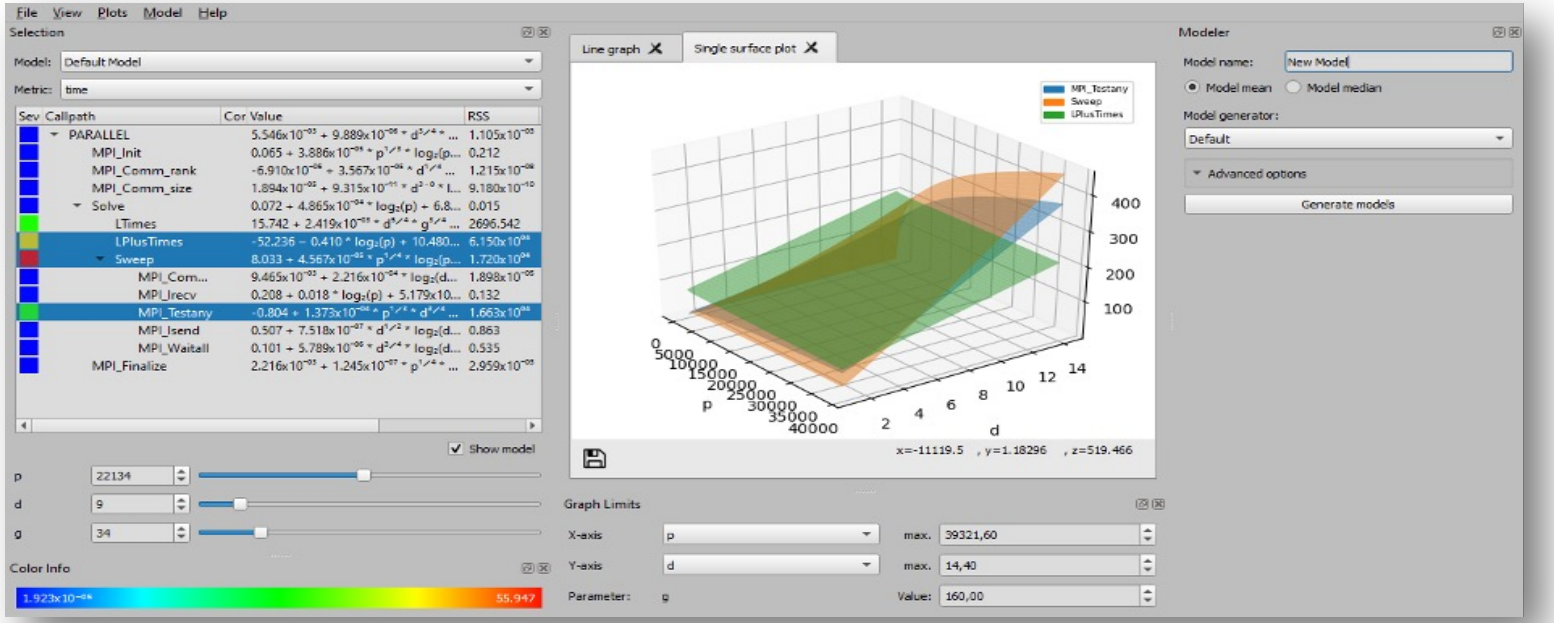
- Identify kernels
  - Incomplete coverage
- Create models
  - Laborious, difficult

Extra-P

Extra-P in a nutshell:

- Automatic performance-modelling tool
- Generates **empirical performance models** that predict the scaling behavior of the different parts of an application
- Supports the user in identifying scalability bugs

Part of the program whose scaling behavior is unintentionally poor



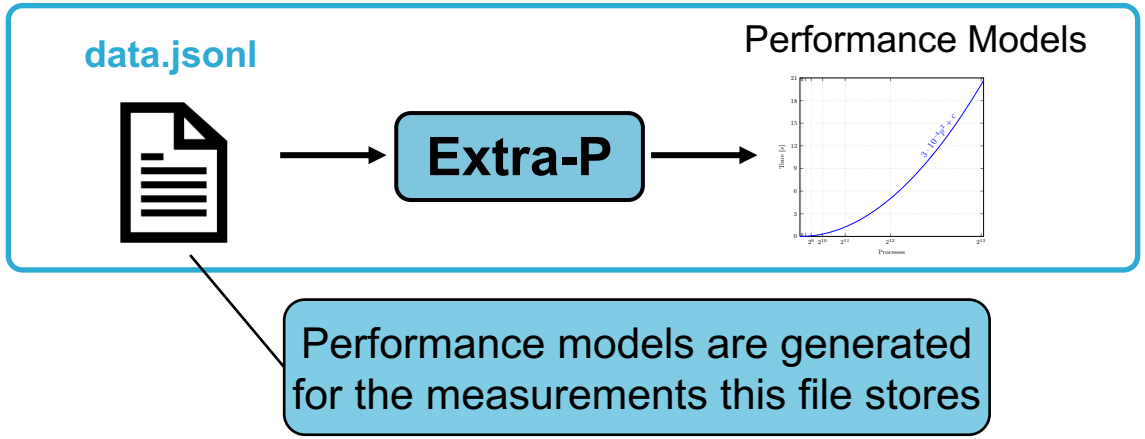
https://github.com/extra-p/extrap

New

YouTube https://youtu.be/Cv2YRCMWqBM

Extra-P in ADMIRE: Continuous Modeling

- In ADMIRE: No human in the loop
  →Use Extra-P **programming interface**
- Whenever **new traces** are available, just **reinvoke** Extra-P



- **Storesfine graded I/O data in the JSONL file**
  → new profiles can be simply appended
- **Model fine graded I/O function up till metrics like total I/O time**

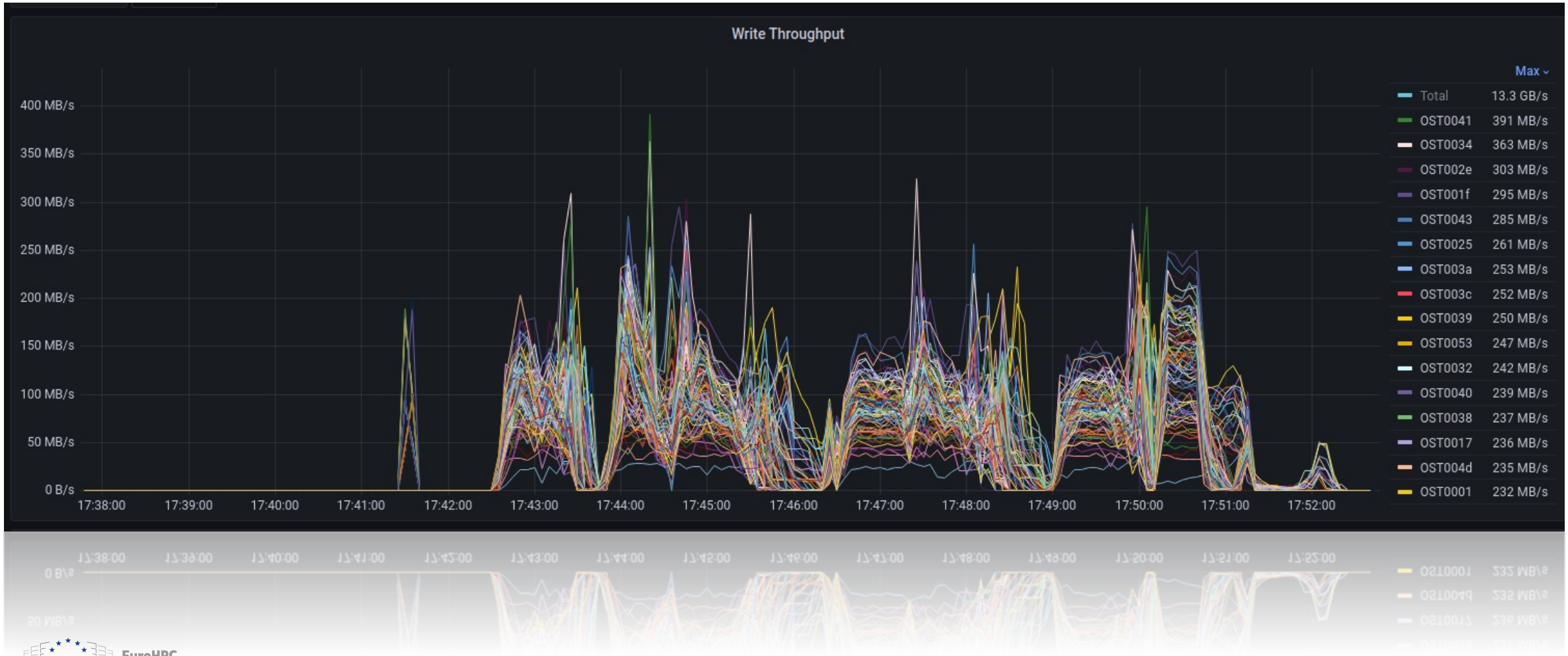
```python
# Imports
from extrap.entities.callpath import Callpath
from extrap.entities.metric import Metric
from extrap.fileio.file_reader.jsonlines_file_reader import read_jsonlines_file
from extrap.modelers.model_generator import ModelGenerator
import numpy as np

# Load the JSONL data. For different file formats, there are
# different read libraries in extrap.fileio
experiment = read_jsonlines_file("./data.jsonl")

# Initialize model generator
model_generator = ModelGenerator(experiment)

# Create models from data
model_generator.model_all()

# ------ At this point, the models are generated
# Next, let's see how to evaluate them

# 1) Specify the metric and call path,e.g., default and root, respectively
cp0 = Callpath("<root>"), Metric("<default>")
pm = model_generator.models[cp0].hypothesis.function
print(f"Model evaluated at x = 5 and y = 5:\n{pm.evaluate([5,5])}\n")

# 2) Or iterate over all of them:
for model in model_generator.models.values():
    # Measurment points
    pts = model.measurements
    print(f"Measurement points are:\n{pts}\n")

    # Evaulated function @ measurement points
    pred = model.predictions
    print(f"Prediction points are:\n{pred}\n")

    # The actual model
    print(f"Model is:\n{model.hypothesis.function}\n")

    # Evaluate at specific points
    # Here, data.jsonl is a 2D data set, hence, a 2d array must be provided
    m = model.hypothesis.function.evaluate(
        np.array([[1, 24, 58], [1, 29, 30]]))
    print(f"Model evaluated at x = [1 24 58] and y = [1 29 30] is:\n{m}\n")

    # Or evaluate just at a single point [x,y]
    pm = model.hypothesis.function.evaluate([2, 3])
    print(f"Model evaluated at x = 2 and y = 3:\n{pm}\n\n")
```

Monitoring Proxy: Application Side

Monitoring Proxy Captures traces at different levels for each application (example: IMB-IO)

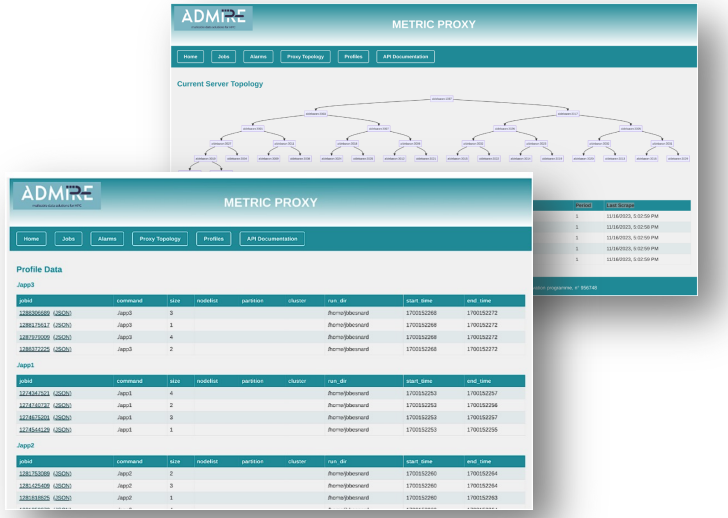
Tracing I/O: File System Side

File system side is also available, but currently not used for performance modeling (Ex: HACC-IO)



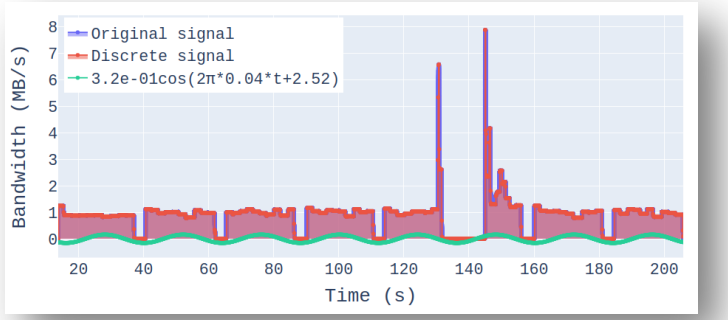
Modeling I/O With Extra-P

- Support for four new tools has been added
  - TMIO (Tracing MPI-IO)
  - Darshan
  - Tau metric proxy
  - FTIO



Metric proxy: Graphical user interface

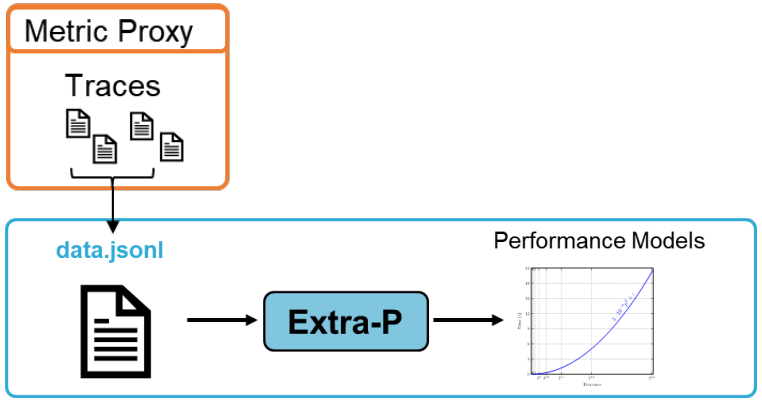
Source	Key Advantage
TMIO	Async I/O, I/O requirements, flexibility
Darshan	Universal support
Metric proxy	Very detailed traces, **on-the-fly modeling**
FTIO	Modeling I/O phases



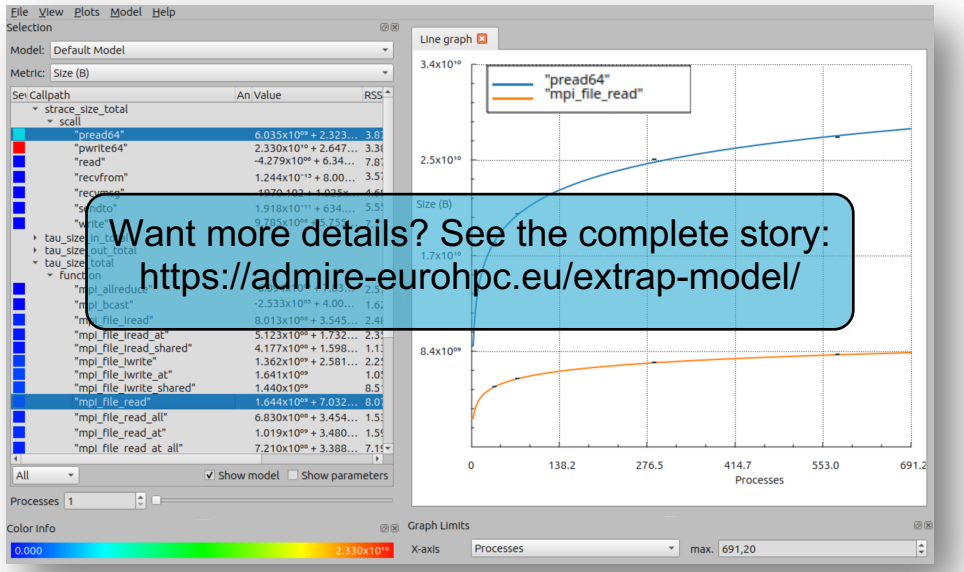
FTIO: Frequency techniques for I/O

Extra-P and Tau Metric Proxy

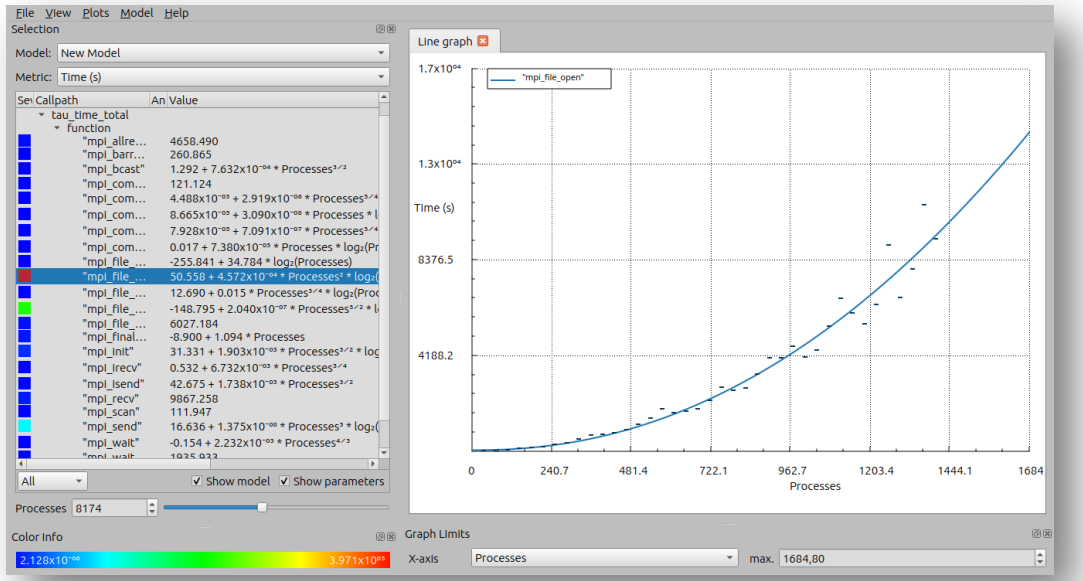
- Metric proxy:
  - Provides traces at different levels
  - Groups traces into a single profile
  - Different methods for grouping exist



IMB-IO

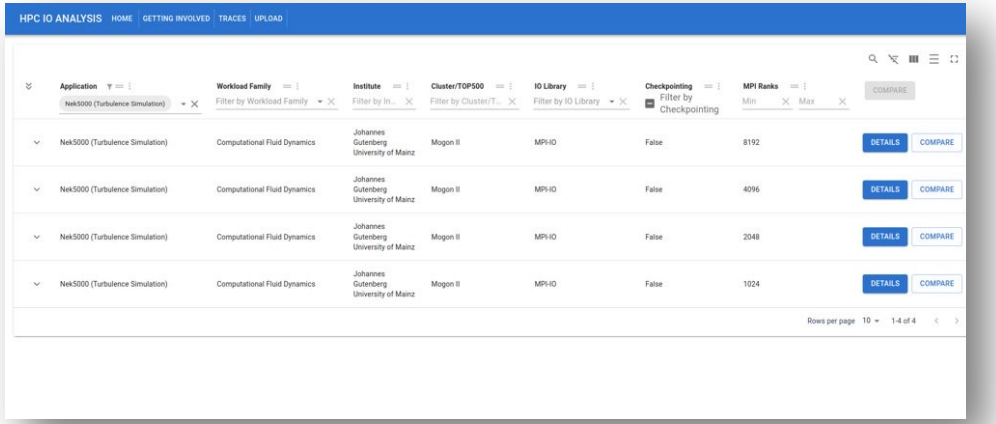


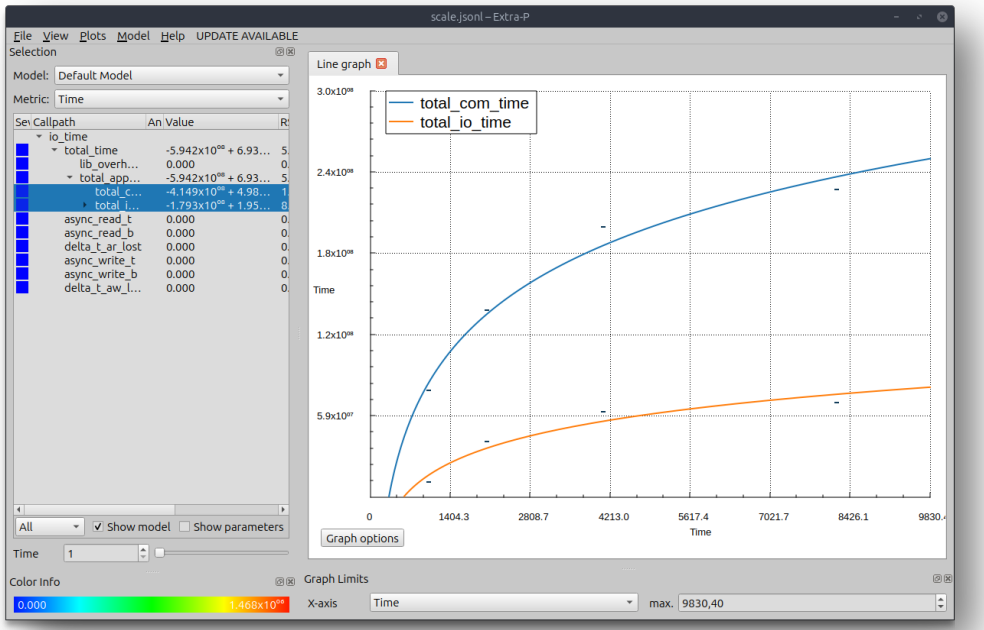
Nek5000



Extra-P and Darshan

- Nek5000 simulation from Mainz trace Website
- Generate a performance model from any Darshan File (with and without heatmap)





Acknowledgment

- We acknowledge the support of the European Commission and the German Federal Ministry of Education and Research (BMBF) under the EuroHPC Programmes DEEP-SEA (GA no. 955606, BMBF funding no. 16HPC015) and ADMIRE (GA no. 956748, BMBF funding no. 16HPC006K), which receive support from the European Union's Horizon 2020 programme and DE, FR, ES, GR, BE, SE, GB, CH (DEEP-SEA) or DE, FR, ES, IT, PL, SE (ADMIRE).

- We gratefully acknowledge the computing time provided on the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt. This is funded by the Federal Ministry of Education and Research, and the state governments participating on the basis of the resolutions of the GWK for national high performance computing at universities *(www.nhr-verein.de/unsere-partner)*. Moreover, we acknowledge the computing time provided on the high-performance computer at the University of Turin from the laboratory on High-Performance Computing for Artificial Intelligence.

Thank you for your attention!

Questions?