Automatic Tuning of HPC Applications for Performance and Energy Efficiency.

Michael Gerndt

Technische Universität München

READEX

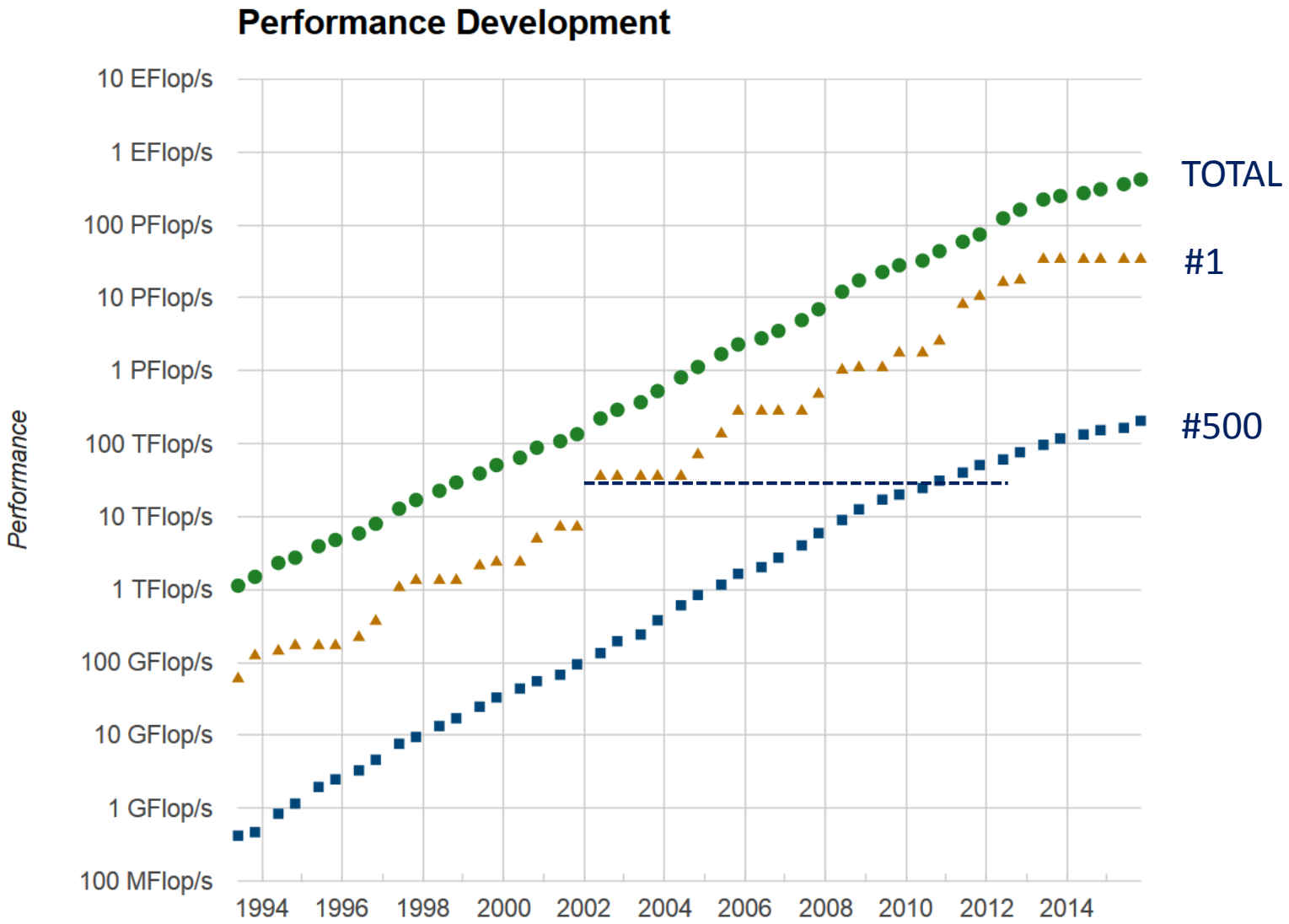
Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing

European Commission

Horizon 2020
European Union funding
for Research & Innovation

SuperMUC: 3 Petaflops ($3*10^{15}$=quadrillion), 3 MW

TOP 500 List


Performance Development
Performance
10 EFlop/s
1 EFlop/s
100 PFlop/s
10 PFlop/s
1 PFlop/s
100 TFlop/s
10 TFlop/s
1 TFlop/s
100 GFlop/s
10 GFlop/s
1 GFlop/s
100 MFlop/s
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
TOTAL
#1
#500

TOP 5 Systems: Linear Extens for Exascale

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)	
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808	*19 = 340 MW
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209	*36 = 302 MW
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890	*50 = 390 MW
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660	*89 = 1115 MW
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945	*100 = 394 MW

Project overview

- READEX

  Runtime **E**xploitation of **A**pplication **D**ynamism for **E**nergy-efficient e**X**ascale Computing

- Starting date:

  1. September 2015

- Duration:

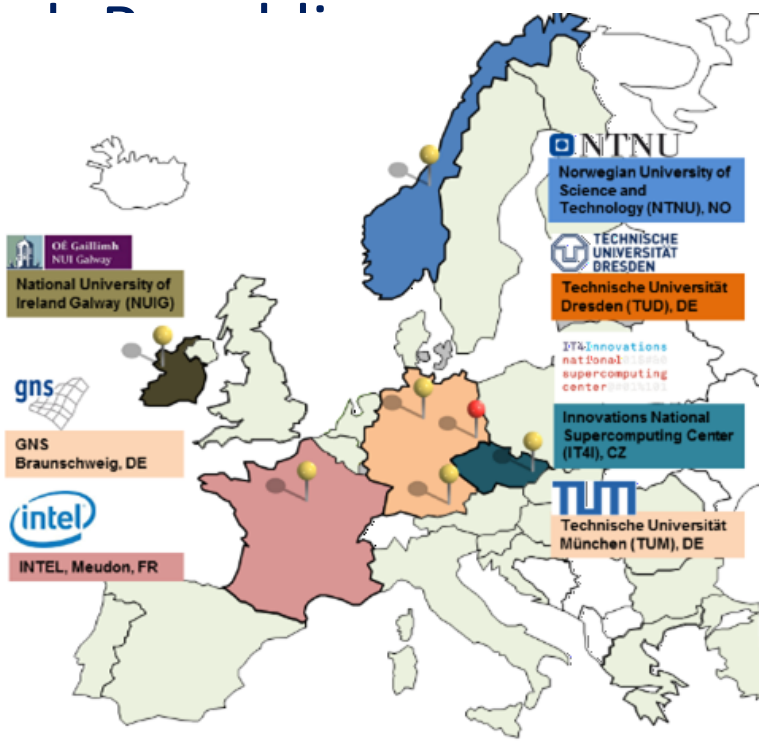
  3 years

- Funding:

  European Commission Horizon 2020 grant agreement 671657

Project partners

- Technische Universität Dresden (Coordinator), Germany
- Norwegian University of Science and Technology, Norway
- Innovations National Supercomputing Center, Cz
- Technische Universität München, Germany
- Intel Exascale Centre, France
- GNS Braunschweig, Germany
- National University of Ireland Galway, Ireland



Motivation

Challenges

- Energy consumption
- Extreme scale
- Dynamism

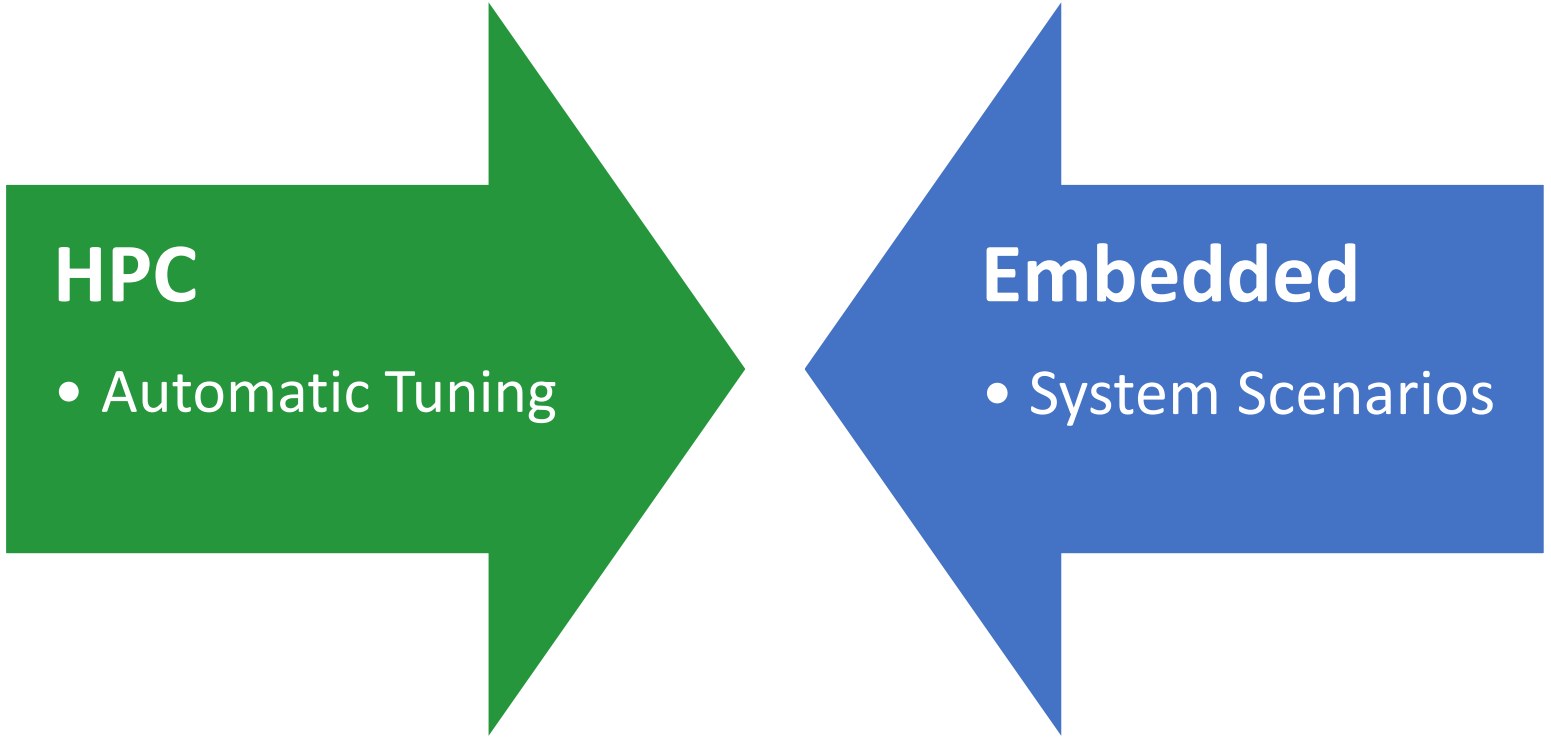
Problems

- Awareness
- Ability
- Effort

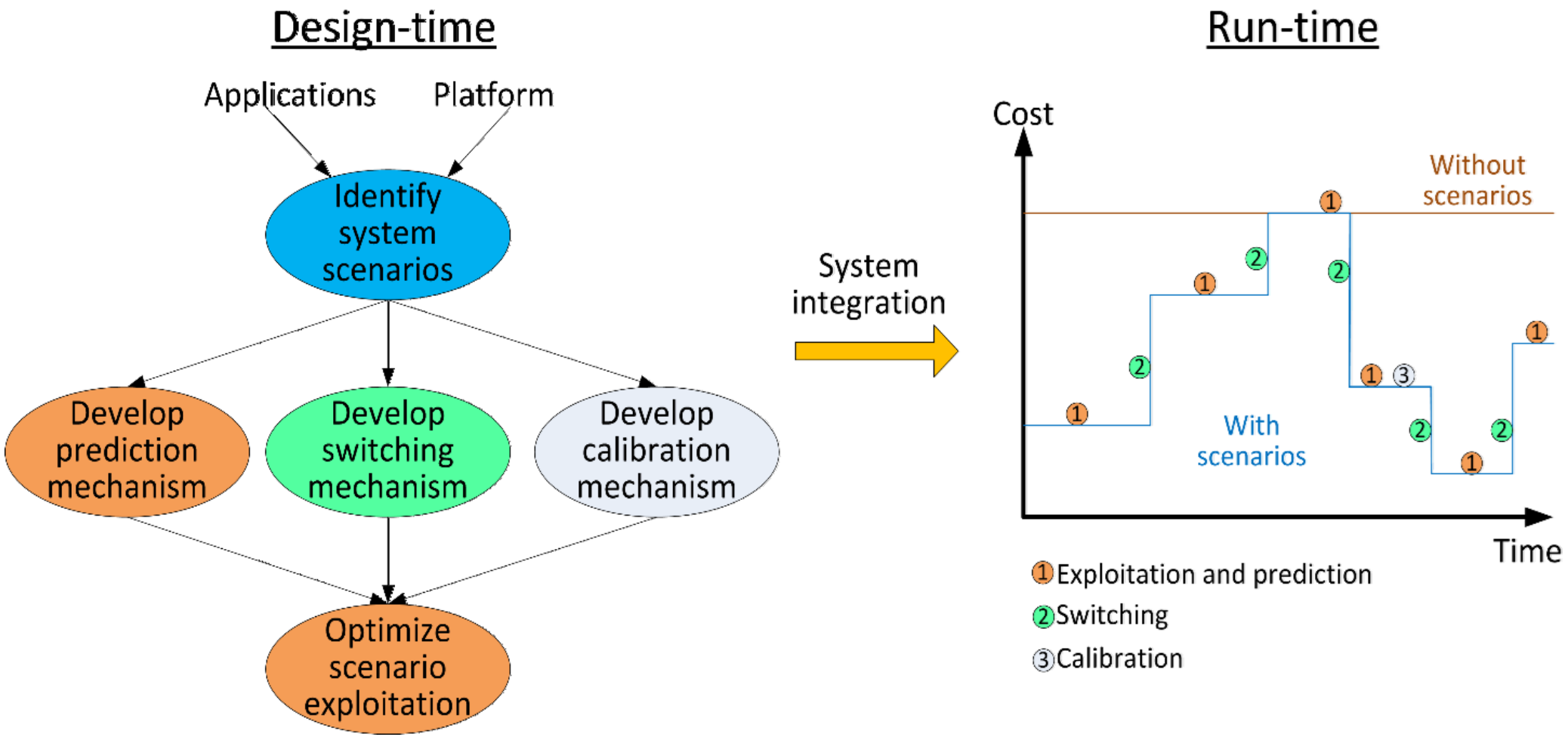
Solution

- Dynamism
- Automatic tuning
- Design-/Run-time

General idea



Systems Scenario based Methodology



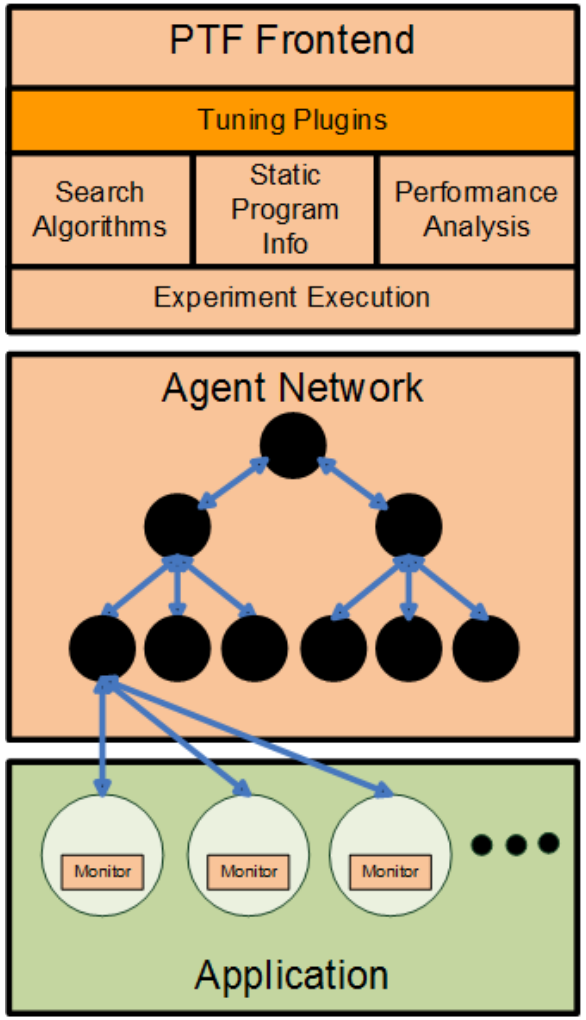
Outline

Static Tuning with the
Periscope Tuning Framework

Dynamic Tuning with the
READEX Tool Suite and Methodology

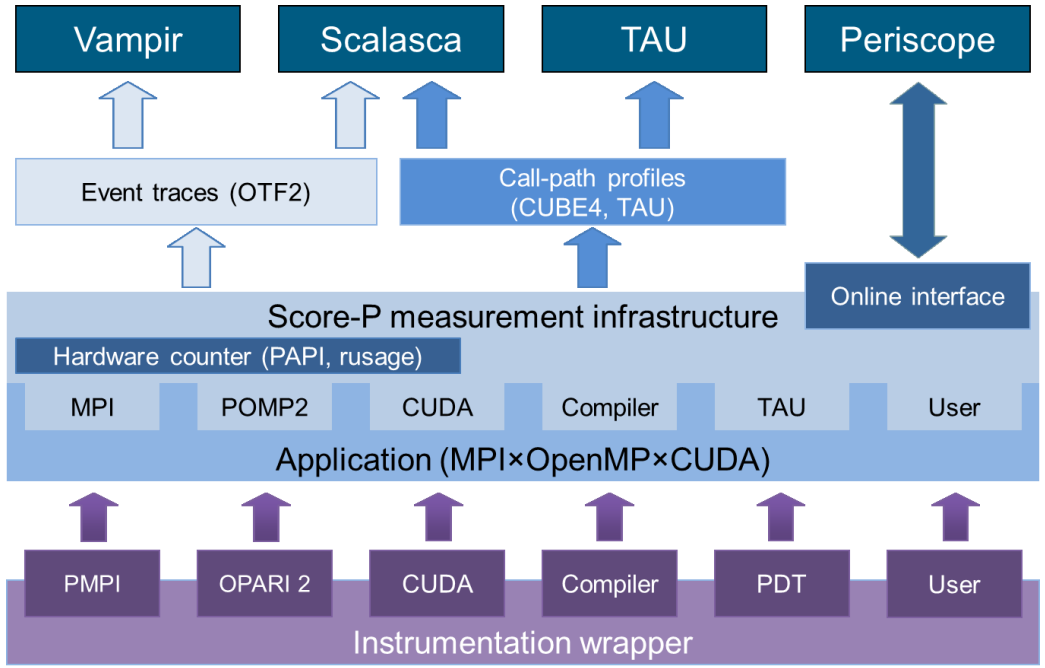
Periscope Tuning Framework

- Automatic application analysis & tuning
  - Tune performance and energy (statically)
  - Plug-in-based architecture
  - Evaluate alternatives online
  - Scalable and distributed framework
- Support variety of parallel paradigms
  - MPI, OpenMP, OpenCL, Parallel pattern
- Developed in the AutoTune EU-FP7 project



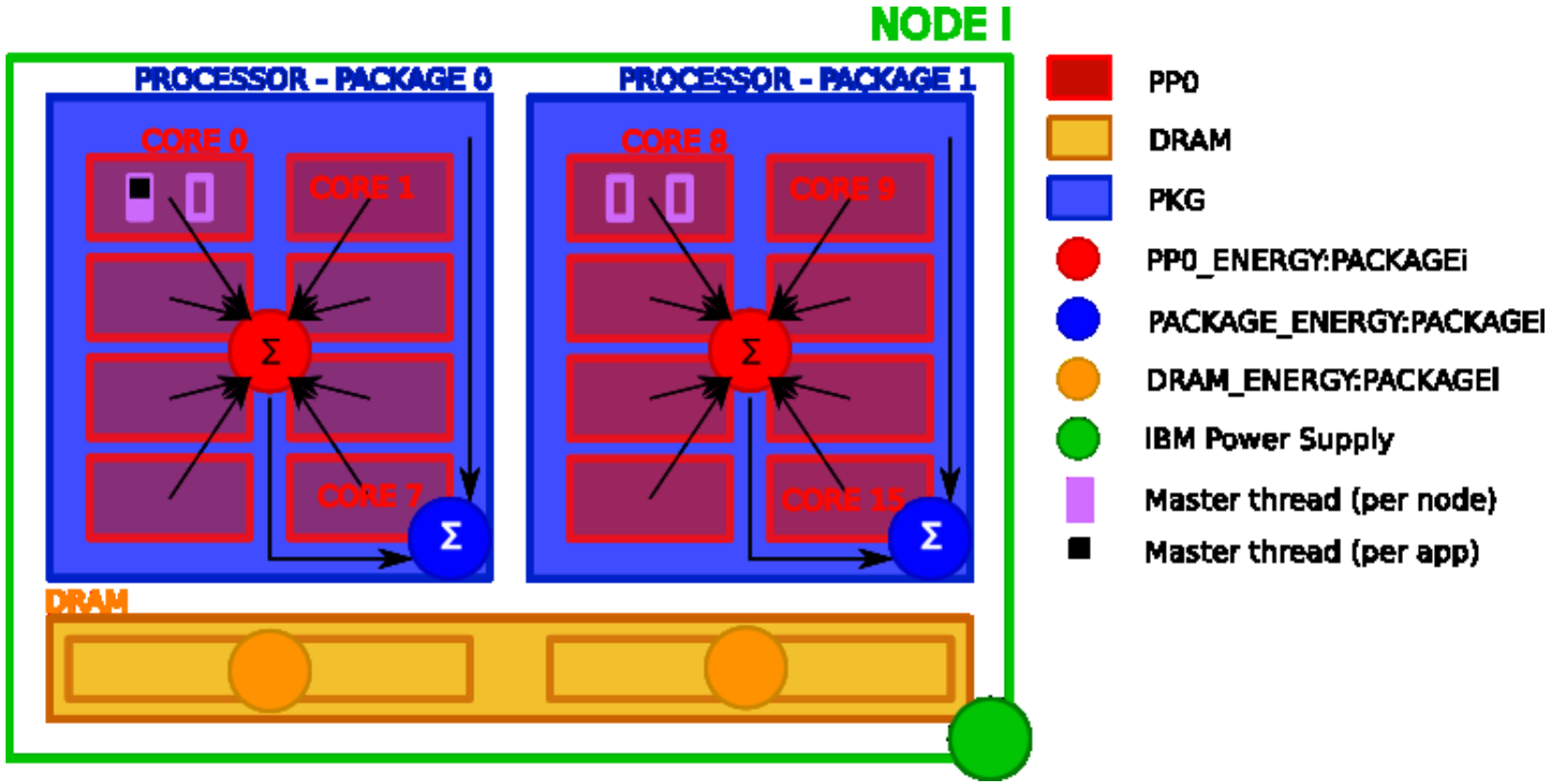
Scalable Performance Measurement Infrastructure for Parallel Codes

Common instrumentation and measurement infrastructure

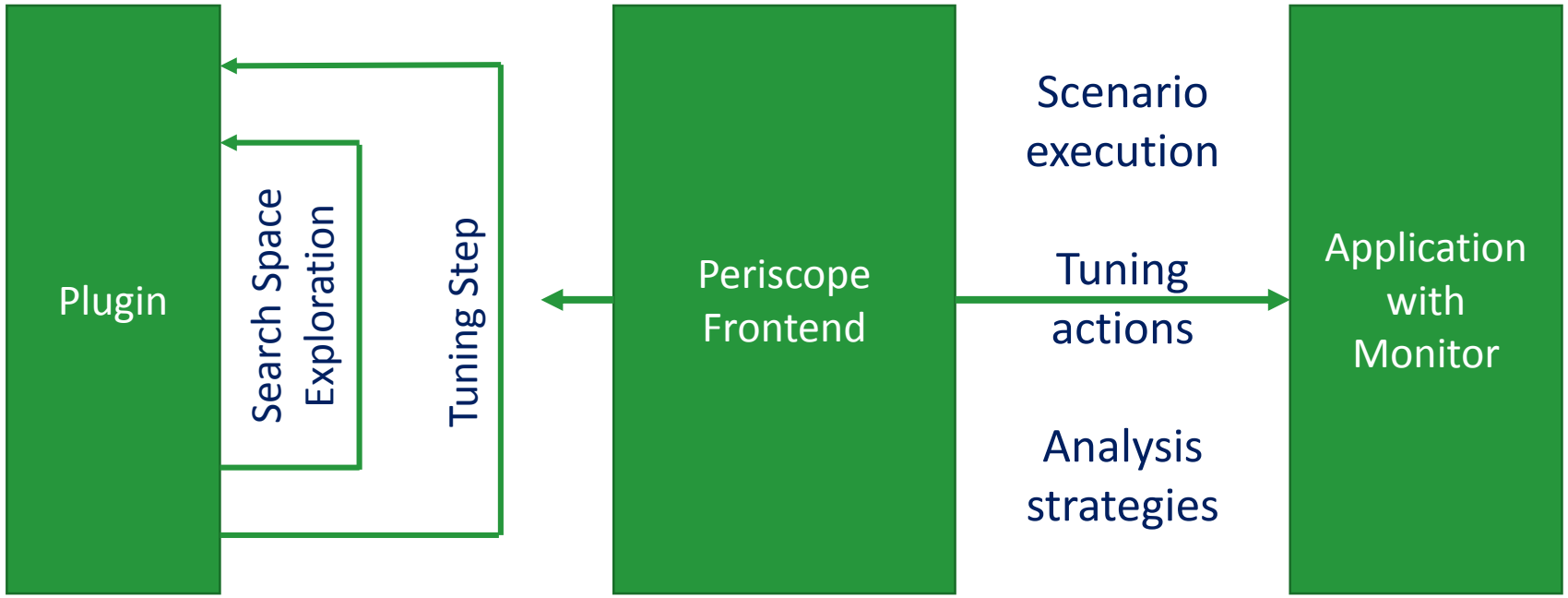


Score-P

ENOPT Library for Energy Measurements



Tuning Plugin Interface



Tuning Plugins

- MPI parameters
  - Eager Limit, Buffer space, collective algorithms
  - Application restart or MPIT Tools Interface
- DVFS
  - Frequency tuning for energy delay product
  - Model-based prediction of frequency
  - Region level tuning
- Parallelism capping
  - Thread number tuning for energy delay product
  - Exhaustive and curve fitting based prediction

European Commission
Horizon 2020
European Union funding for Research & Innovation

Tuning Plugins

- Master/worker
  - Partition factor and number of workers
  - Prediction through performance model based on data measured in pre-analysis
- Parallel Pattern
  - Tuning replication and buffers between pipeline stages
  - Based on component distribution via StarPU
- OpenCL tuning
  - Compiler flags for offline compilation
  - NDRange tuning

European Commission
Horizon 2020
European Union funding for Research & Innovation

Tuning Plugins

- MPI IO
  - Tuning data sieving and number of aggregators
  - Exhaustive and model based
- Compiler Flag Selection
  - Automatic recompilation and execution
  - Selective recompilation based on pre-analysis
  - Exhaustive and individual search
  - Scenario analysis for significant routines
  - Combination with Pathway

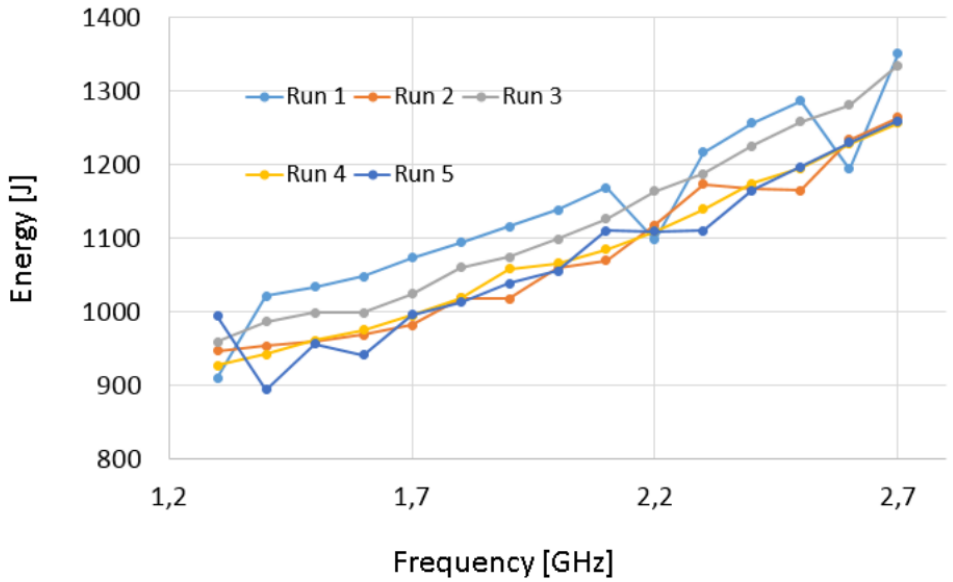
European Commission
Horizon 2020
European Union funding for Research & Innovation

Plugin Evaluation

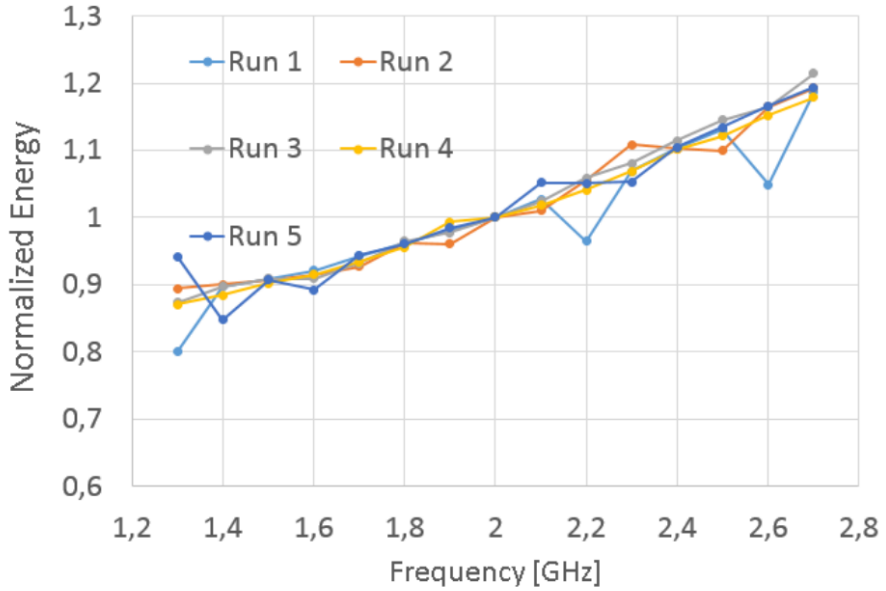
Application Name	CFS	DVFS	MPI Parameters	Patterns	Master Worker
APEX-MAP	✗	✓	✗	✗	✗
BLAST	✗	✗	✗	✗	✓
Convolution	✓	✗	✗	✗	✗
FaceDetect	✗	✗	✗	✓	✗
FSSIM	✓	✓	✓	✗	✗
HydroC	✓	✗	✗	✗	✗
NPB	✓	✓	✓	✗	✗
pmatmul	✓	✓	✓	✗	✗
SeisSol	✓	✓	✓	✗	✗
Sip	✓	✓	✗	✗	✗
S2F2M	✓	✗	✓	✗	✓
Model_primes	✓	✗	✓	✗	✗
nw	✗	✗	✗	✓	✗

European Commission | Horizon 2020 European Union funding for Research & Innovation

Variation of Energy Measurements

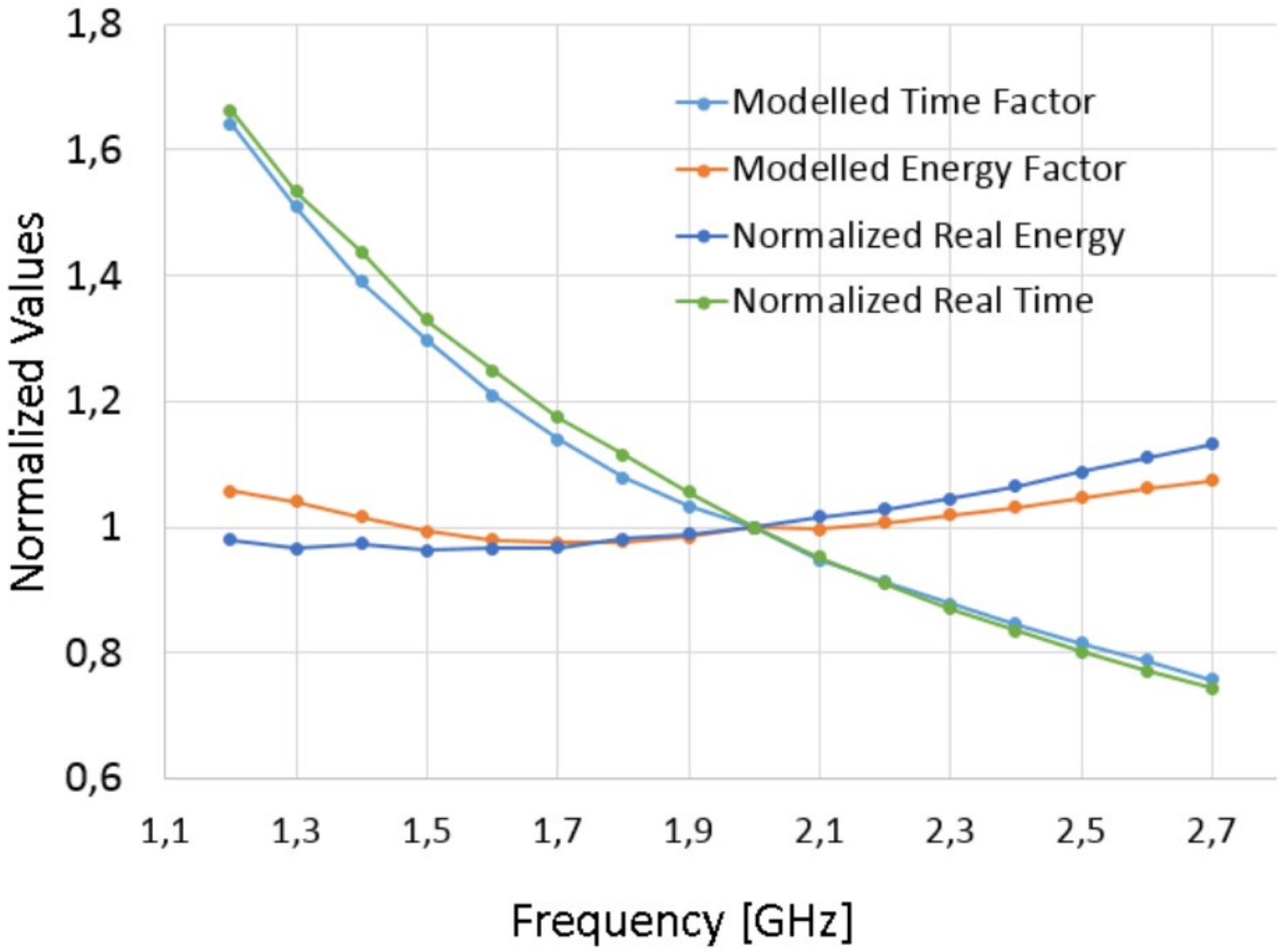


Energy consumption of the SeisSol application at different compute nodes.

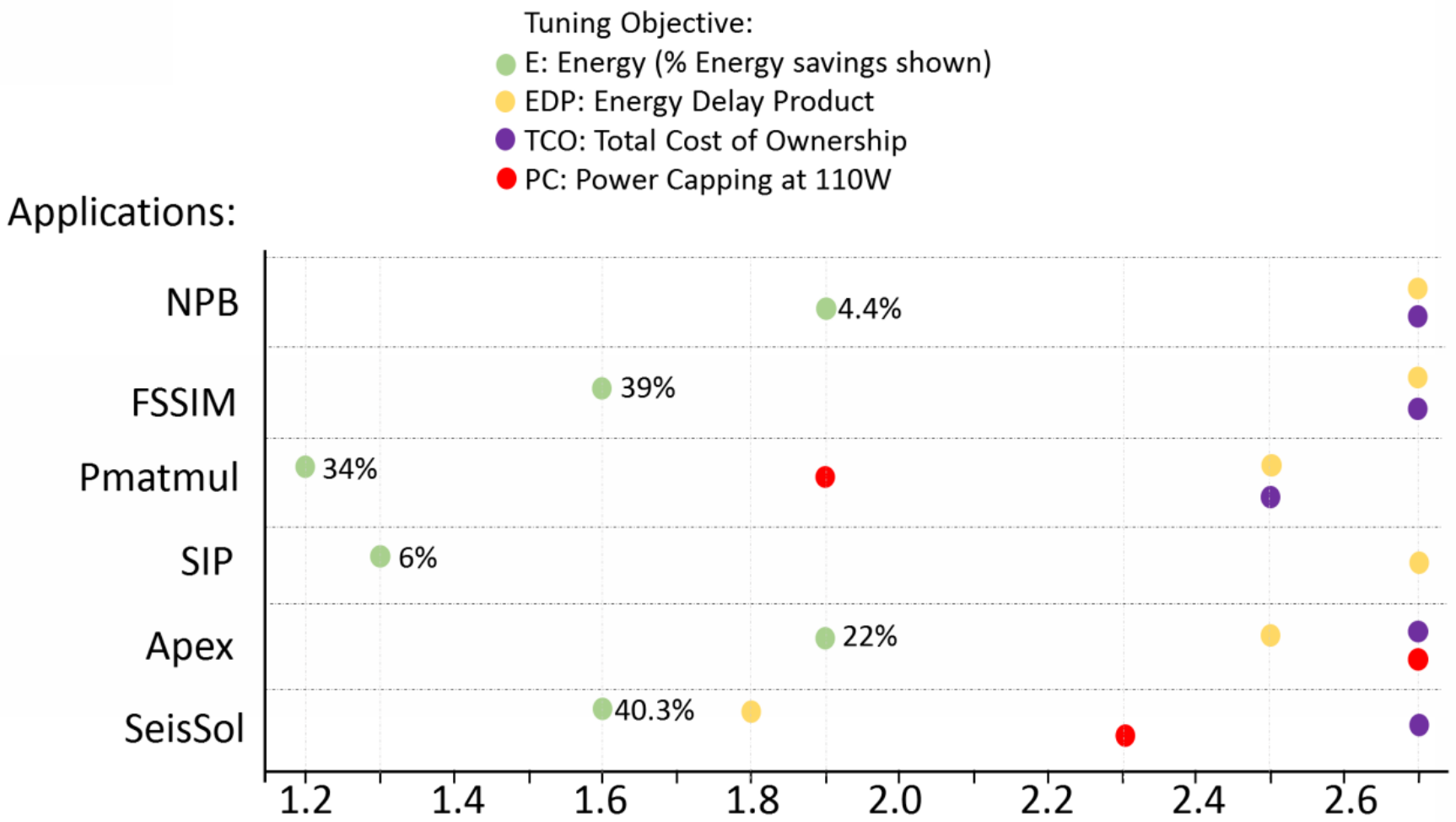


Normalized energy consumption of the SeisSol application at different compute

Predicted vs Measured Time for Seissol



Tuning with the Persicope Tuning Framework



Application Dynamism: Beyond Static Tuning

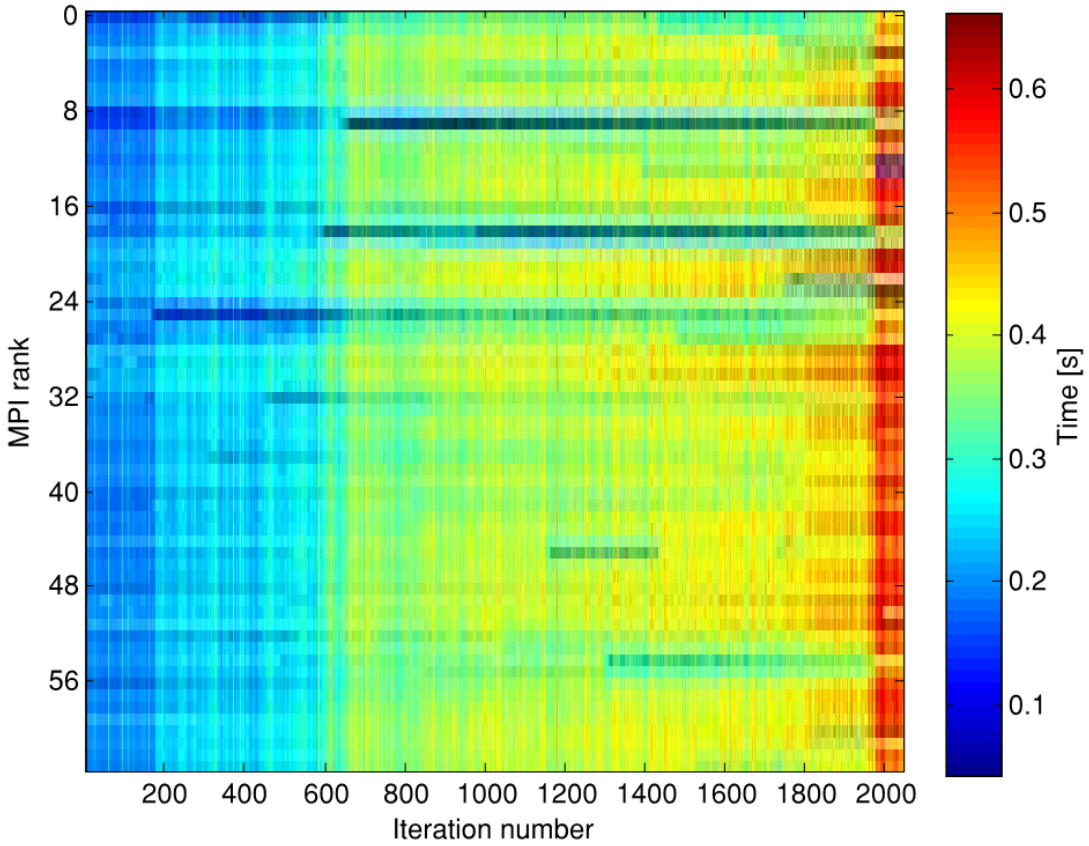
```
int main(void) {

  // Initialize application
  // Initialize experiment variables

  int num_iterations = 2;

  for (int iter = 1; iter <= num_iterations; iter++) {
    // Start phase region
    // Read PhaseCharct
    laplace3D(); // significant region
    residue = reduction(); // insignificant region
    fftw_execute(); // significant region
    // End phase region
  }

  // Post-processing:
  // Write noise matrices to disk for visualization
  // Terminate application

  MPI_Finalize();
  return 0;
}
```

One iteration of the progress loop

Horizon 2020
European Union funding
for Research & Innovation

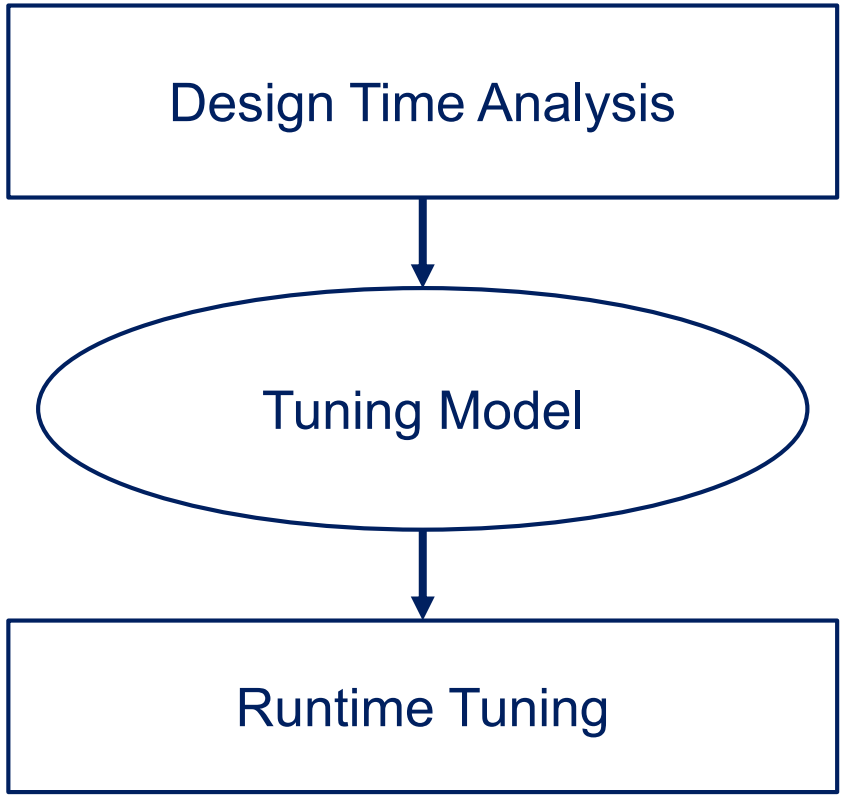
Inter-phase Dynamism

All-to-all Performance
2048 phases



PEPC Benchmark of the DEISA Benchmark Suite

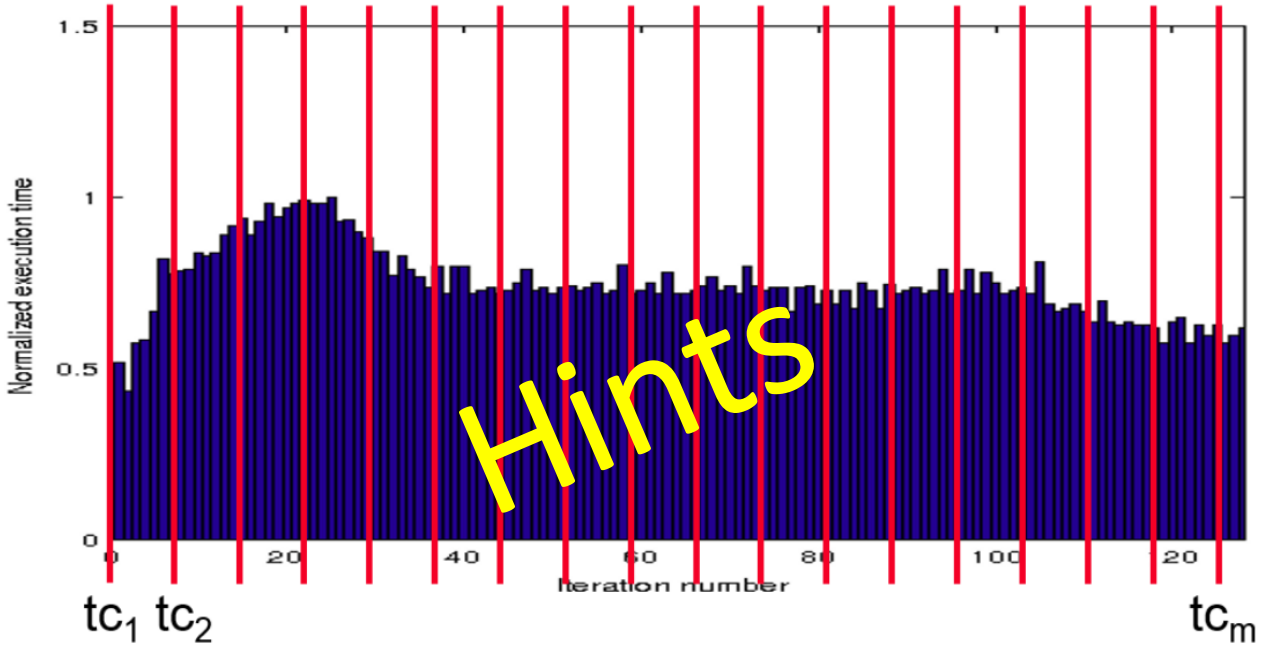
Scenario-Based Tuning



Periscope Tuning Framework (PTF)

READEX Runtime Library (RRL)

Design Time Analysis

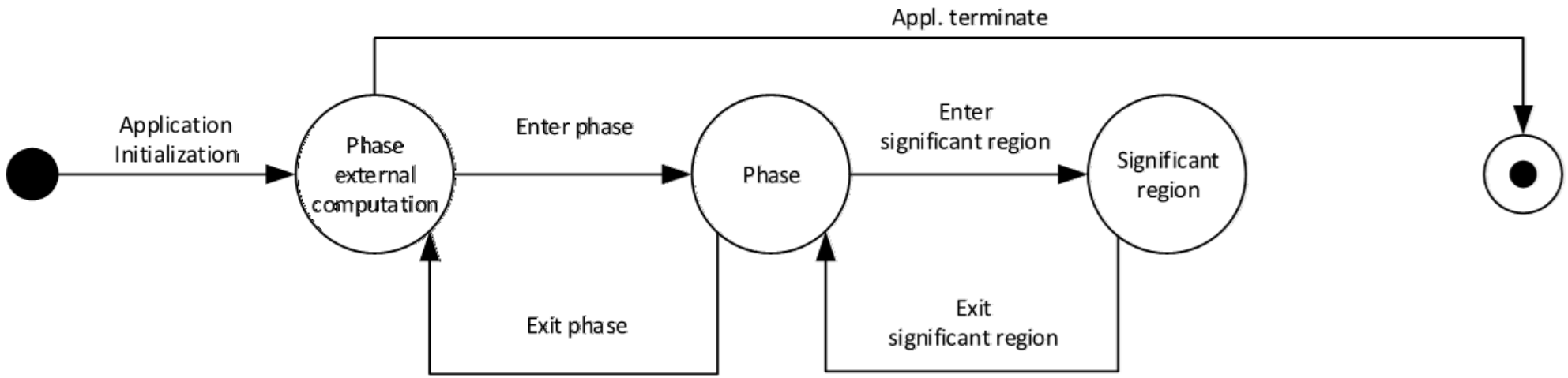


Tuning Model

- Scenarios: set of runtime situations (rts)
- Classifiers: RTS → S
- Selector: Context → CFG

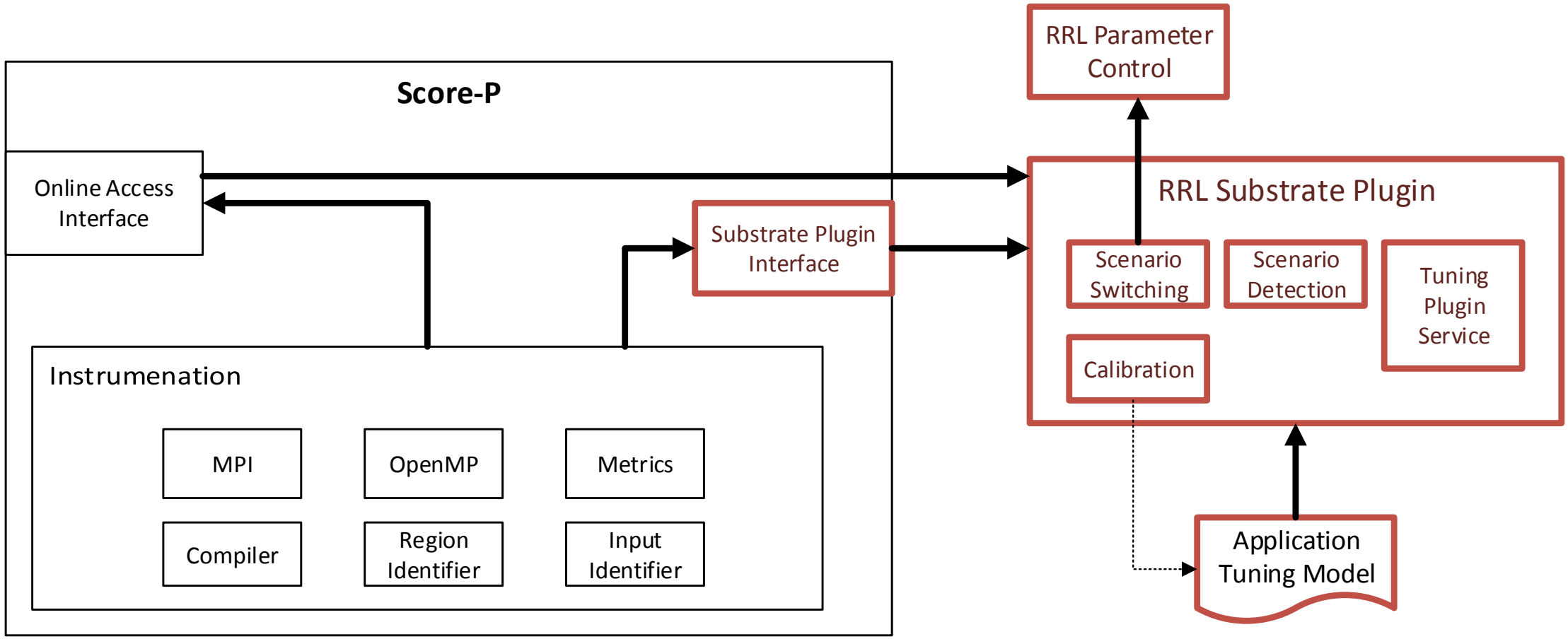
- Tuning cylces
  - Captures intra-phase dynamism
  - Creates phase TM
- Sequence of tuning cycles
  - Captures inter-phase dynamism
  - Creates inter-phase TM
- DTA for multiple inputs
  - Captures input dynamism
  - Creates application TM

Runtime Tuning with the READEX Runtime Library



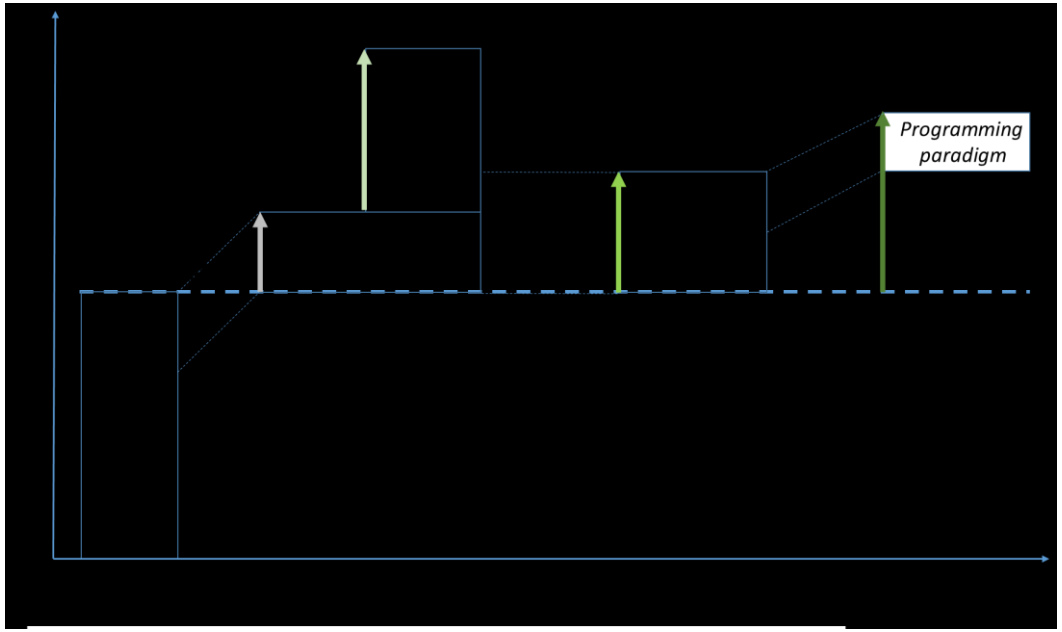
Enter phase:	Capture phase identifiers
Enter significant region:	Classify rts; apply selector; perform switching
Exit significant region:	Save objective value
Exit phase:	Perform calibration

RRL Architecture



Validation and project goals

- **Goal: Validate the effect of READEX using real-world applications**
  - Co-design process:
    - Hand-tune selected applications
    - Compare results with automatic static and dynamic tuning
  - Energy measurements using HDEEM infrastructure



Conclusion

- Energy-efficiency at exascale
  - Application developers and users will have to care
- Lack of capabilities
  - Awareness
  - Expertise
  - Resources
- Proposed solution – READEX:
  - Exploit dynamism
  - Detect at design time, exploit at run-time
  - Tools-aided autotuning methodology

Thank you! Questions?