READEX

Yury Oleynik

Technische Universität München

READEX
Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing

European Commission
Horizon 2020
European Union funding
for Research & Innovation

Outline

- Project overview
- Motivation
- Previous works
- Approach
- Conclusion

Project overview

- READEX

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

- Starting date:

  1. September 2015

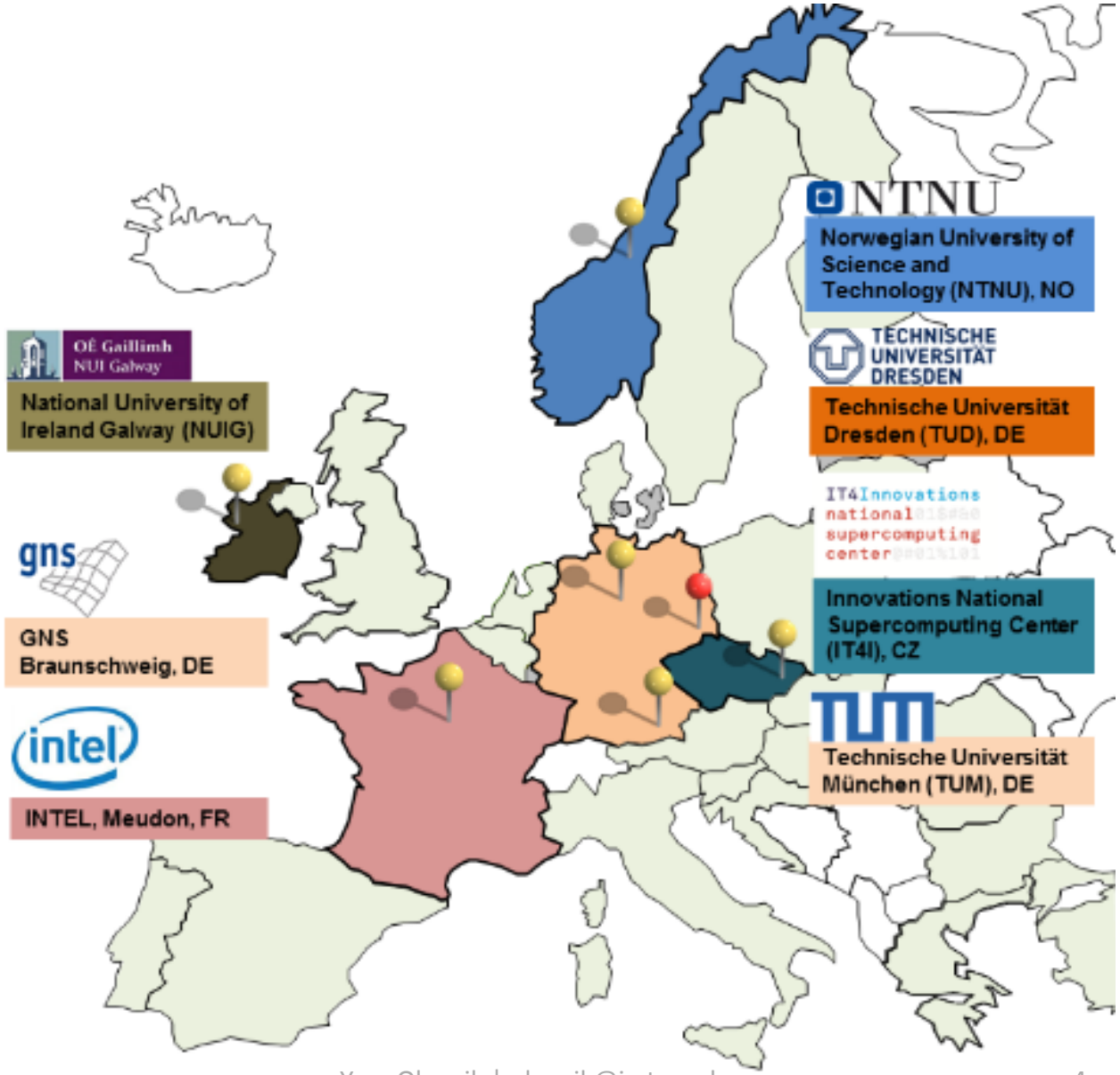
- Duration:

  3 years

- Funding:

  European Commission Horizon 2020 grant agreement 671657

Project partners



READEX
Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing

European Commission
Horizon 2020
European Union funding for Research & Innovation

Motivation

Challenges

- Energy consumption
- Extreme scale
- Dynamism

Problems

- Awareness
- Ability
- Effort

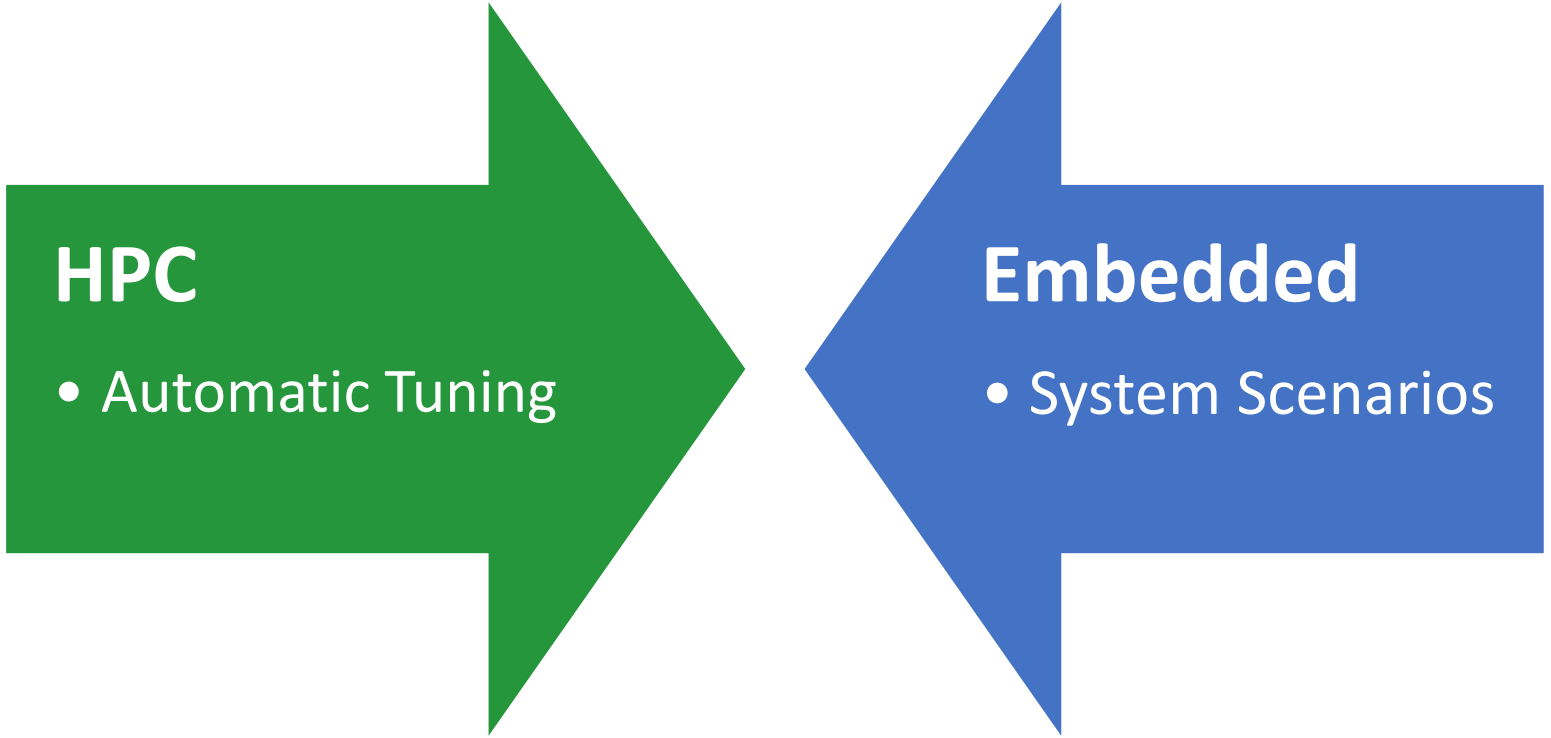
Solution

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

Yury Oleynik | oleynik@in.tum.de

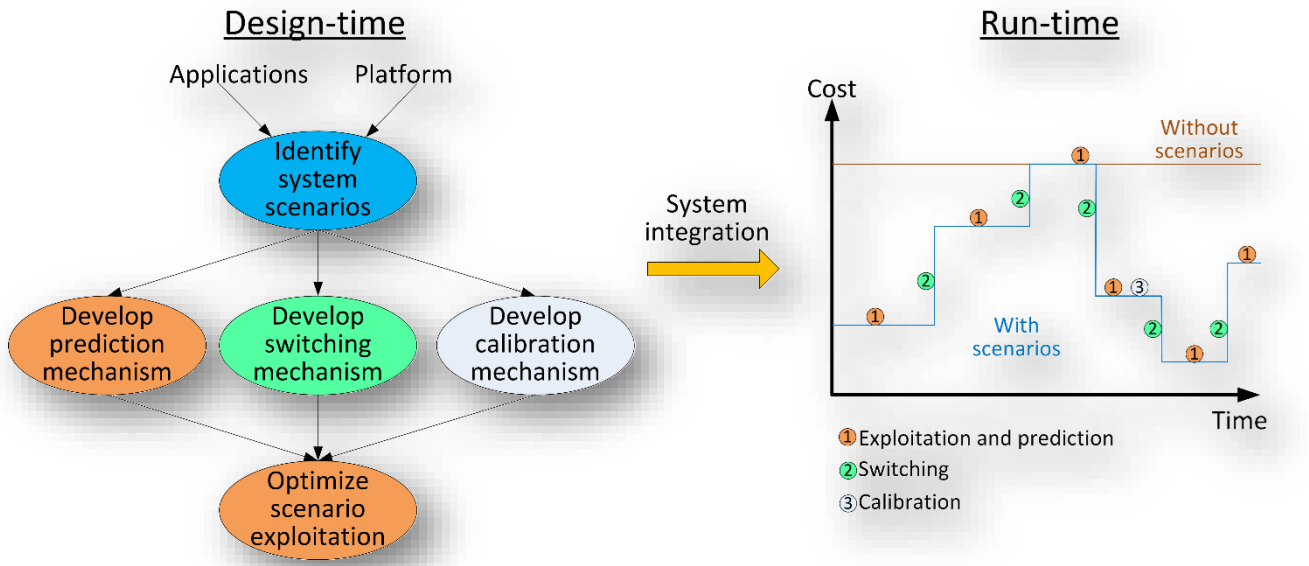
European Commission
Horizon 2020
European Union funding for Research & Innovation

General idea



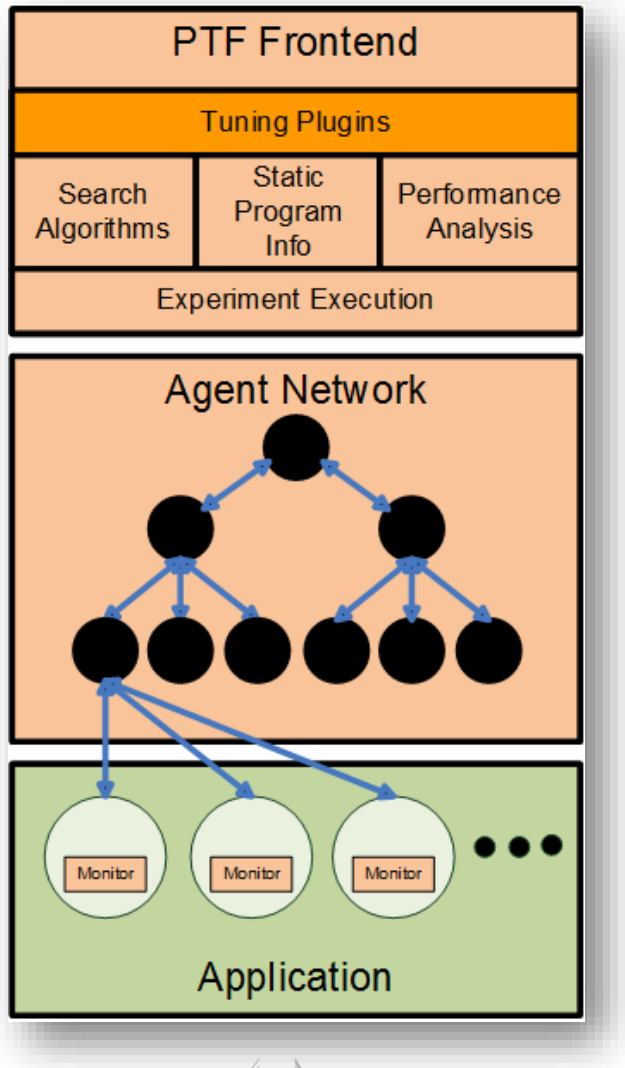
Systems scenarios

- **System Scenario based Methodology**
  - Formalism for dynamic auto-tuning in the embedded systems world
  - Detect and analyze dynamism in applications at design-time
  - Switch parameters at run-time based on detected scenarios



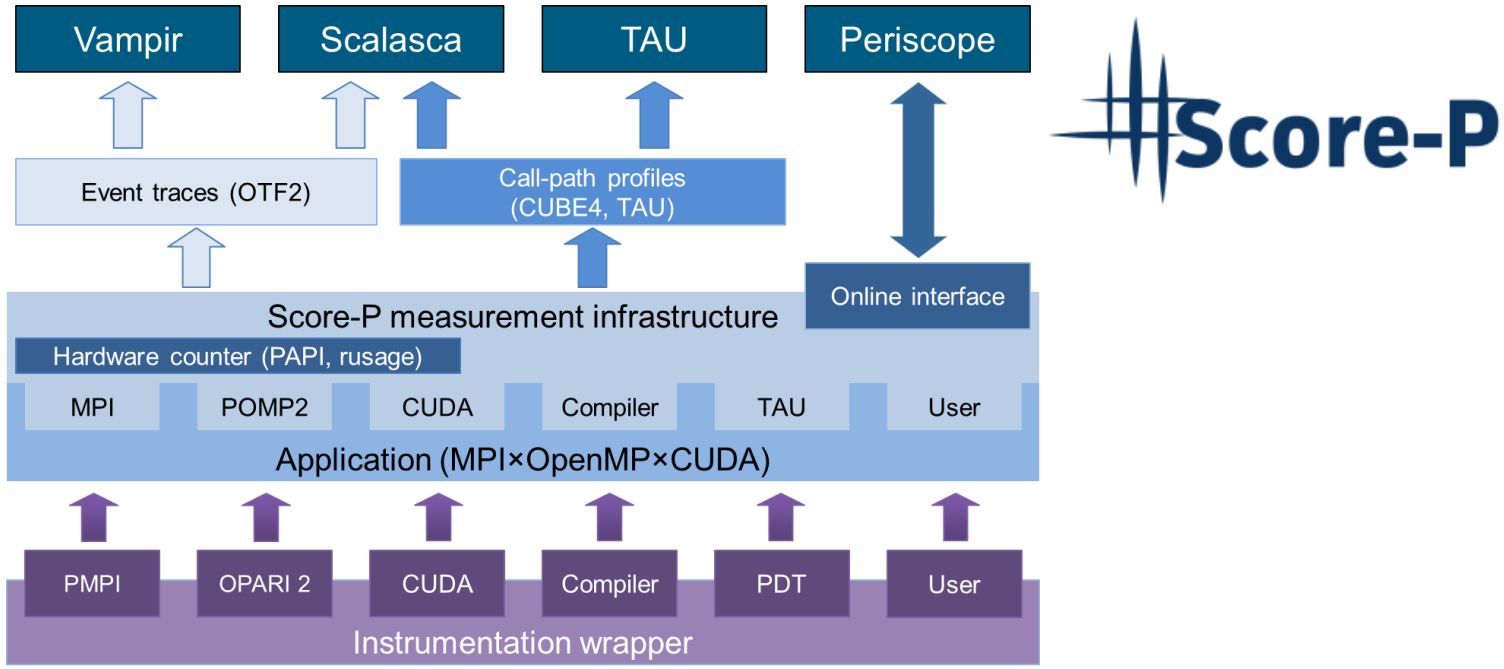
Periscope Tuning Framework

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

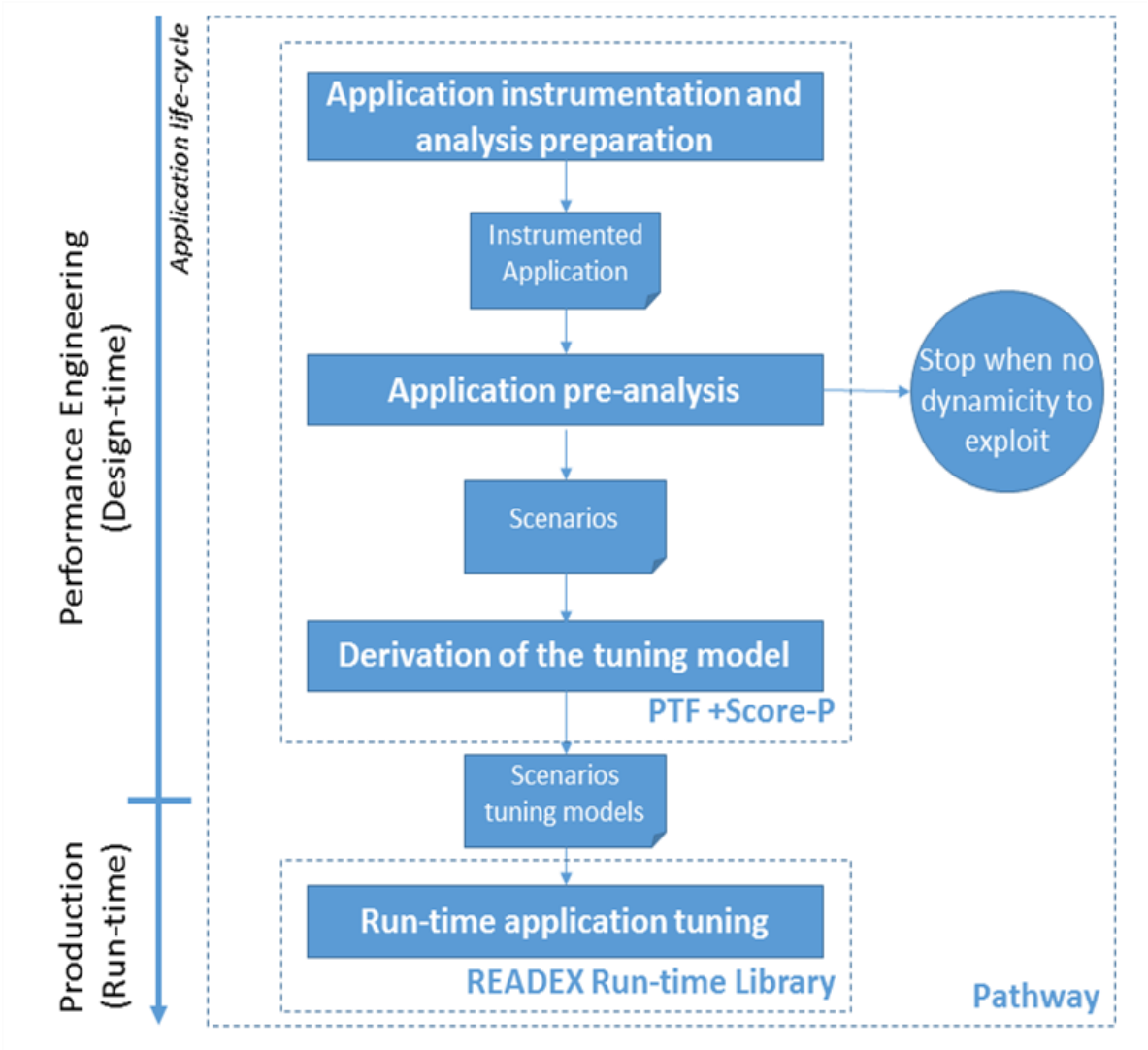


Score-P

- **Scalable Performance Measurement Infrastructure for Parallel Codes**
  - Common instrumentation and measurement infrastructure

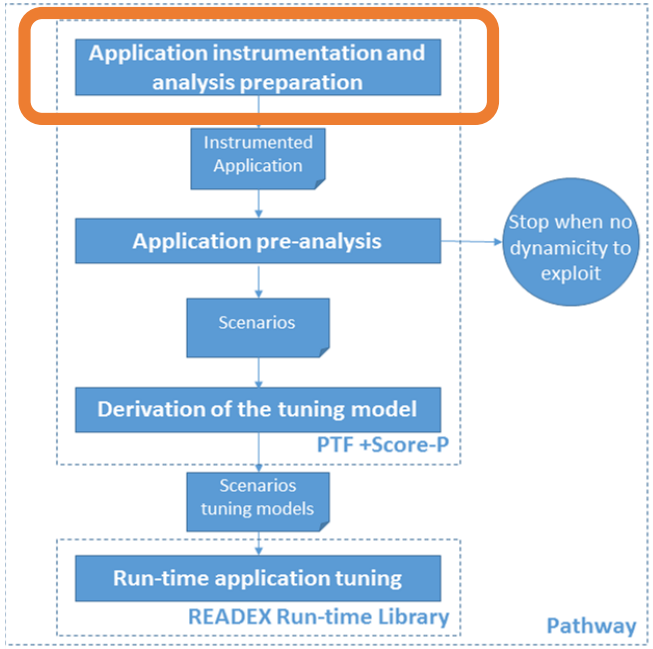


Approach


Application life-cycle
Performance Engineering
(Design-time)
Production
(Run-time)
Application instrumentation and analysis preparation
Instrumented Application
Application pre-analysis
Stop when no dynamicity to exploit
Scenarios
Derivation of the tuning model
PTF +Score-P
Scenarios tuning models
Run-time application tuning
READEX Run-time Library
Pathway

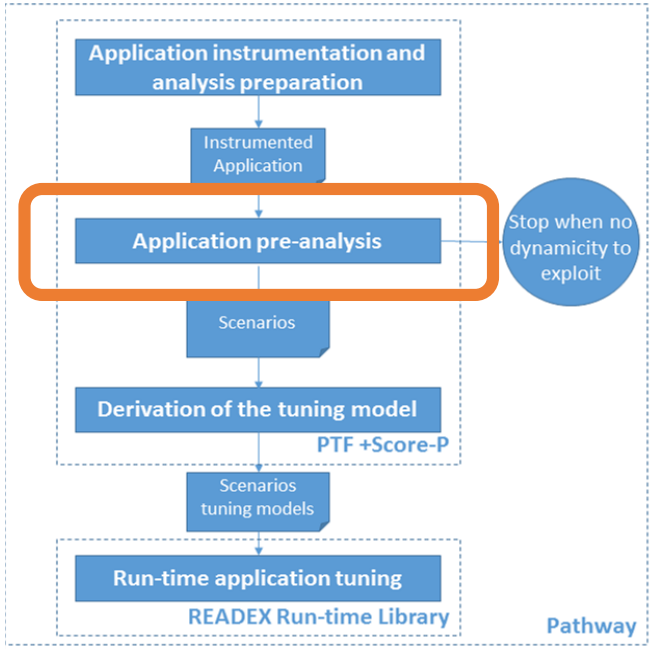
READEX: Instrumentation and Preparation

- **Goal: Insert probes to analyse application dynamism**
  - **Score-P** instrumentation infrastructure
    - Compiler/manual region instrumentation
    - MPI, CUDA, OpenMP, …
  - **READEX User API** to expose
    - Application identifier and
    - Application-level tuning parameter



READEX: Application Pre-Analysis

- **Goal: Find exploitable dynamism**
  - Record energy and hardware counter per region
  - Determine differences between regions
    - Different **Run-time Situations** (RTS)
    - RTS == point in a multi-dimensional space
  - **System Scenarios:**
    - Group RTS into similarity clusters
    - Points with similar performance and energy properties
  - Scenarios recognized based on **Identifier**
    - Detect scenario before it is executed
    - Examples: function call-path, function arguments values, …

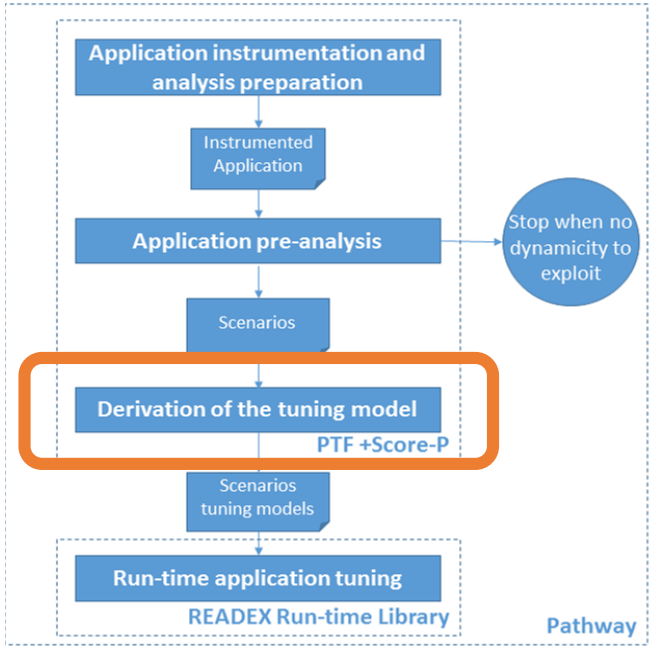


Yury Oleynik | oleynik@in.tum.de

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

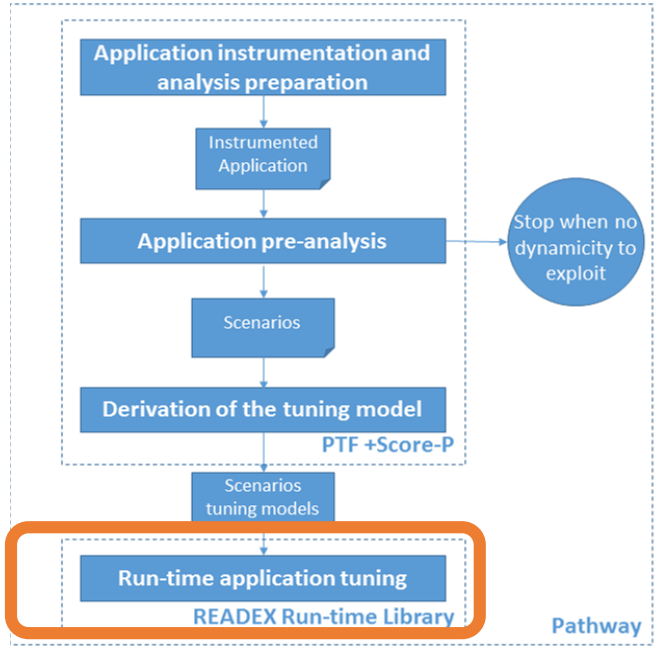
READEX: Tuning Model Derivation

- **Goal: Create a tuning model**
- Find Pareto-optimal tuning parameters for detected scenarios
  - Based on Objective Funtion
  - Use PTF tuning plugin infrastructure
  - Perform multiple application runs



READEX: Run-time Application Tuning

- **Goal: Perform efficient configuration switching**
  - Low-overhead READEX Run-time Library (RRL)
    - Based on Score-P measurement infrastructure
    - Detect upcoming scenarios and apply configuration change if beneficial
  - Create new configurations for unknown RTS
    - Calibration

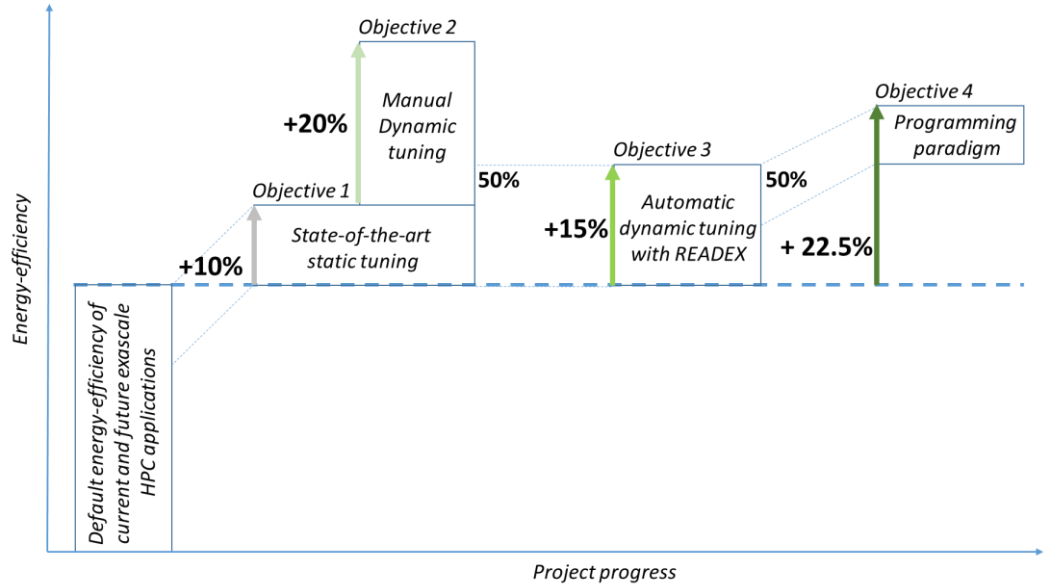


Yury Oleynik | oleynik@in.tum.de

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

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-, exploit at run-time
  - Tools-aided auto-tuning methodology

Thank you! Questions?