<table>
<thead>
<tr>
<th>ANTAREX</th>
<th>AutoTuning and Adaptivity appRoach for Energy efficient eXascale HPC systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call:</td>
<td>H2020-FET-HPC-1-2014</td>
</tr>
<tr>
<td>Type of action:</td>
<td>H2020: Research &amp; Innovation Actions (RIA)</td>
</tr>
<tr>
<td>Topics:</td>
<td>HPC Core Technologies, Programming Environments and Algorithms for Extreme Parallelism and Extreme Data Applications</td>
</tr>
<tr>
<td>Subtopic</td>
<td>b) Programming methodologies, environments, languages and tools</td>
</tr>
<tr>
<td>Project Coordinator</td>
<td>Cristina Silvano, Politecnico di Milano</td>
</tr>
<tr>
<td>EC Contribution</td>
<td>3, 115, 251 euro</td>
</tr>
<tr>
<td>Project start:</td>
<td>September 1st, 2015 (duration 3 years) Kick-off meeting held at CINECA (Italy)</td>
</tr>
</tbody>
</table>
Target Scenario

• To reach the DARPA’s target of **20MW** of Exascale supercomputers projected to 2020, current supercomputers must achieve an energy efficiency “quantum leap”, pushing towards a goal of **50 GFlops/W**.

• Heterogeneous systems currently dominate the top of the **Green500 list** and this dominance is expected to be a trend for the next coming years to reach the target of 20MW Exascale supercomputers.

• Energy-efficient heterogeneous supercomputers need to be coupled with a radically new software stack capable of exploiting the benefits offered by heterogeneity at all the different levels (supercomputer, job, node) to meet the scalability and energy efficiency required by the Exascale era.
Target Scenario

To reach the DARPA’s target of 20MW of Exascale supercomputers projected to 2020, current supercomputers must achieve an energy efficiency “quantum leap” pushing towards a goal of 50 GFlops/W.

Heterogeneous systems currently dominate the top of the Green500 list and this dominance is expected to be a trend for the next coming years to reach the target of 20MW Exascale supercomputers.

To fulfill this target, energy-efficient heterogeneous supercomputers need to be coupled with a radically new software stack capable of exploiting the benefits offered by heterogeneity at all the different levels (supercomputer, job, node) to meet the scalability and energy efficiency required by the Exascale era.

The main goal of the ANTAREX project is to provide a breakthrough approach to express by a Domain Specific Language the application self-adaptivity and to runtime manage and autotune applications for green and heterogeneous High Performance Computing (HPC) systems up to the Exascale level.
ANTAREX Main Objectives

1. **Dynamic self-monitoring and self-adaptivity or «autotuning»** HPC applications with respect to changing workloads, operating conditions and computing resources.

2. **Programming models and languages to express self-adaptivity and extra-functional properties.** Introducing a separation of concerns between extra-functional (self-adaptivity, parallelisation, energy/thermal management) strategies and application functionality by the design of a new aspect-oriented **Domain Specific Language**.

3. **Exploiting heterogeneous computing resources in Green HPC platforms by runtime resource and power management**.
The ANTAREX Approach

- Introducing a new DSL (as and extension of the LARA language) for expressing **adaptivity and autotuning strategies**.
- Enabling performance/energy control capabilities by introducing **software knobs** (including application parameters, code transformations and code variants).
- Designing optimal control-loops to manage **performance/energy control knobs** (job dispatching, resource management and dynamic voltage/frequency scaling) at different time scale (compile time, deployment time and run-time).
- Monitoring the evolution of the supercomputer as well as the **application status and requirements** to provide this information to the ANTAREX energy/performance-aware software stack.
Compiling and Optimizing with LARA DSL: Motivations

- Enable sophisticated strategies for code instrumentation and synthesis/compiler optimizations
- Enable users and DSE engines to fully explore compiler optimizations
- Enable users to apply the most suitable compiler sequence according to code and target architecture
- Enable more advanced control than the ones using pragmas/directives/switches
- Enable an unified view of design-flows
LARA approach: design benefits

Reusable Strategies

Custom Targetability

Design Exploration

app 1 + aspect = design 1

app 2 + aspect = design 2

app N + aspect = design N

app + aspect 1 = design 1

app + aspect 2 = design 2

app + aspect N = design N

original app + aspect = design 1

original app + aspect = design 2

original app + aspect = design N

design parameters
LARA is a domain-specific language (DSL) inspired by aspect-oriented programming concepts.

LARA has been designed as much agnostic as possible to:
- Target architecture
- Target (host) programming language
- Target toolchain

LARA highlights:
- Secondary concerns detached from application logic code
- Useful to program strategies for instrumentation and synthesis/compiler optimizations
- Fully explore compiler optimizations and optimization sequences, according to code and target architectures
- Provides an unified view and DSE mechanisms

**LARA**

**Aspect specification**

**Action specification**

**Attribute specification**

**Language Specification**

**LARA Aspects**

**Application Source Code**

**weaver**

**larac**

**Aspect-IR**

**Woven Application**

---

LARA for Runtime Adaptivity

• Extending LARA with native support for runtime adaptivity strategies:
  – Separation of concerns: Functional and non-functional descriptions decoupled
  – LARA apply sections migrated to runtime

• LARA Approach to Design-Flows
  – Express and capture transformation schemes and strategies
  – Controlled and guided through aspects and strategies powered by LARA
  – Program monitoring, profiling
  – Program code transformations, sequences of compiler optimizations
  – Program mapping strategies, including hardware/software partitioning
Application Autotuning
Application Autotuning (1)

• Define mechanisms and strategies to be adopted at application-level to on-line adapt the application behaviour and the platform configuration.
• To use the ANTAREX DSL to instrument the application for monitoring and to support the autotuning.
• To search for the best combination of the software knobs (i.e. application parameters, code transformations and code variants) w.r.t. the execution environment.
• Given the huge search space, the autotuner will search the set of software knobs that are expected to impact the performance and energy efficiency of the application by using on-line learning techniques.
Application Autotuning (2)

- The framework will implement a **collect-analyse-decide-act loop** to make the application behaviour self-aware.

- **Collect** phase implemented by the monitoring framework.
- **Analyse** phase uses data coming from the monitoring phase to build a model between the application design space (software knobs) to high-level metrics (such as performance and energy).
- Results of **analyse** phase used during the **decide** phase for selecting the most suitable combination of application parameters and code transformations -- given the allocated resources.
- **Act** phase implements in the application the decisions taken according to the constraints assigned by the resource and power manager.
Progress beyond state-of-the-art

Application Scenarios

Self-Adaptivity & Application Autotuning

Runtime Resource & Power Management

Target HPC Systems

Programming Models, Language & Compilers
Two HPC application scenarios

1. A biopharmaceutical HPC application for accelerating drug discovery provided by DOMPE and to be deployed on the 1.2 PetaFlops heterogeneous Ne XtScale Tier-1 Intel-based IBM system at CINECA.

2. A self-adaptive navigation system provided by SYGIC to be used in smart cities and deployed on the server-side on an heterogeneous Intel-based PetaFlops class system given by IT4Innovations Supercomputing Center.

- Performance metrics extracted from the two use cases will be modelled to extrapolate these results towards Exascale systems expected by the end of 2020.
- These use cases have been selected due to their significance in emerging application trends and thus by their direct economic exploitability and relevant social impact.
Use Case 1: Computer Accelerated Drug Discovery System

- **Personalized Medicine** will enable to “treat the right patient with the right drug at the right dose at the right time.” <FDA>

- **Need of HPC in Drug Discovery**
  - HPC Molecular Simulations
LiGen Project: A High Performance Workflow for Chemistry Driven de Novo Design
Use Case 2: Self-adaptive Navigation System
Sygic Company develops world`s most popular offline navigation application

and provides professional navigation software for business solutions
Sygic navigation in UC2 context

- Many drivers – many routing requests to HPC system
- Smart City Challenge – serve all city drivers’ requests with global best
http://www.antarex-project.eu/