The ANTAREX Approach to AutoTuning and Adaptivity for Energy efficient HPC systems

The ANTAREX Team

Nesus Fifth Working Group Meeting
Ljubljana, July 8th, 2016
<table>
<thead>
<tr>
<th><strong>ANTAREX</strong></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)</td>
</tr>
</tbody>
</table>
Target Scenario

- To reach the DARPA’s target of 20MW of Exascale supercomputers projected to 2023, 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 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 2023, 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 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-adaptivity** or «autotuning» HPC applications with respect to changing workloads, operating conditions and computing resources.

2. Providing **programming models** and **languages** to express self-adaptivity and non-functional properties.
   
   Developing a new **aspect-oriented Domain Specific Language** enabling the **separation of concerns** between functional and non-functional descriptions and strategies.

3. **Monitoring** the evolution of green HPC platforms and exploiting heterogeneous computing resources by **runtime resource** and **power management**
Application Autotuning

- One or more application parameters, code transformations, and code variants *(application knobs)* can be tuned at runtime.

- **Adaptivity** to adjust the application behavior to the changing operating conditions, usage contexts and resource availability.
Application Autotuning

- One or more application parameters, code transformations and code variants (application knobs) can be tuned at runtime
- Adaptivity to adjust the application behavior to the changing operating conditions, usage contexts and resource availability
- Approximate computing: output just needs to be “good enough” trading off accuracy and throughput
ANTAREX Application Autotuning: Software Knobs

Application Space

Target Independent Space

Target Dependent Space (e.g. Intel Xeon, Intel Xeon Phi)

Application Parameters

Source to Source Code Transformations

Compiler Flags

App

.bin
Exposing knobs at application-level

Metacode of the tunable application:

initialize $M$
initialize $L$

for($\text{angle}_a = 0; \text{angle}_a < 360; \text{angle}_a += 1$) 
  for($\text{angle}_b = 0; \text{angle}_b < 360; \text{angle}_b += 1$) 
  {
    rotate ($L$, $\text{angle}_a$, $\text{angle}_b$)
    overlap ($M$, $L$)
    
    distance ($M$)
    /* more lines of codes */
  }

/* more lines of codes */
Exposing knobs at application-level

Metacode of the tunable application:

initialize M
initialize L

for(angle_a = 0; angle_a < 360; angle_a += 1 + skip_factor1 )
for(angle_b = 0; angle_b < 360; angle_b += 1 + skip_factor2 ) {
    rotate (L, angle_a, angle_b)
    overlap Tunable (M, L, Knob1, Knob2)
    if (overlap > threshold)
        distance Tunable (M, Knob1, Knob2)
    /* more lines of codes */
}
ANTAREX Application Autotuning Loop

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

![Diagram showing the autotuning loop with steps: Collect, Analyse, Decide, Act.](image-url)
The LARA Language

- Enable separation of concerns: non-functional concerns (performance, energy, monitors,...) are decoupled from the application code (functional description).
- Useful to express strategies for instrumentation and synthesis/compiler optimizations
- Fully explore compiler optimization sequences according to code and target architectures
- Support Design Space Exploration mechanisms to fully explore compiler optimizations.
- Enable more advanced control than using pragmas/directives/switches

LARA Design Benefits

Reusable Strategies

Custom Targetability

Design Exploration
LARA-based Tool Flow

Application (C, Java, MATLAB) → Compiler Toolset → Code Output → Analysis Output

Aspects / Strategies (LARA) → Library of Aspects / Strategies
LARA-based Tool Flow

Application

void
filter_subband(float
z[512], float s[32], float
m[32][64]) {
    ...
    for (i=0;i<32;i++) {
        s[i]= 0;
        for (j=0;j<64;j++) {
            s[i]+=m[i][j]*y[j];
        }
    }
    ...
}

Compiler Toolset

Aspects and Strategies

aspectdef monitor1
    select function{}.var{"s"} end
    apply
        insert.after %{if([[var.usage]] >= 10)
            printf("Warning: value >= 10!\n");}%
    end
    condition $var.is_write end
end

Code Output

for (i=0;i<32;i++) {
    s[i]= 0;
    if(s[i] >= 10) printf("Warning: value >= 10!\n");
    for (j=0;j<64;j++) {
        s[i] += m[i][j] * y[j];
        if(s[i] >= 10) printf("Warning: value >= 10!\n");
    }
}

LARA Action: Code Instrumentation
Loop Unrolling: Aspects and Strategies

LARA: “recipes” for compiler optimizations

aspectdef LoopUnroll
    select loop end
    apply
        if($loop.num_iterations <= 32) {
            $loop.exec Unroll(0);
        } else {
            $loop.exec Unroll(2);
        }
    end
    condition
        $loop.is_innermost &&
        $loop.type=="for"
    end
end
LARA: Runtime Adaptivity Dimensions

- **Apply dynamic choices** from algorithm parameters to compiler and mapping optimizations
- **Forms of runtime adaptivity** include:
  - Modifications of *application parameters* (attributes)
  - Selection among *different algorithms* for solving the same problem
  - Different *compiler optimizations* for the same algorithm
  - *Runtime strategies* for partitioning and for mapping computations targeting hardware accelerators
  - *Runtime management* of system resources

Extending LARA with native support for runtime adaptivity strategies
ANTAREX Runtime Framework

Compile Time

Job Scheduler Allocator

Deploy Time

Run Time

AutoTuner
Collector
Power Manager
Offline Compiler
Online Compiler
Hardware Monitor
Application Monitor

ANTAREX Runtime Framework

[Diagram showing the ANTAREX runtime framework with components and processes]
Use Case 1: HPC 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
- **Huge exploration space**
  - Prediction of properties of protein-ligand complexes
  - Verification of synthetic feasibility
- **Massive parallelism but ...**
  - Unpredictable imbalances in computation
  - Dynamic load balancing is critical
Use Case 2: Self-adaptive Navigation System

- Exploit synergies between client-side and server-side: Many drivers – many routing requests to HPC system
- Smart City Challenge: Serve all city drivers’ requests with global best under variable workload

Sygic Company develops world’s most popular offline navigation application & provides professional navigation software for business solutions
Kick-off Meeting held at CINECA (Italy), Sept. 2015

http://www.antarex-project.eu/