Simplify and optimize heterogeneity

Clara Pezuela
CWS HIPEAC, 30 October 2018, Heraklion (Greece)
Simplify & Optimize Hardware Heterogeneity

Simplifying the way developers approach the development of next-generation applications based in heterogeneous hardware architectures, configurations and software systems including heterogeneous clusters, chips and programmable logic devices.

Optimizing various dimensions of software design and operations (energy, performance and security, among others)
Need to design more flexible software abstractions and improved system architectures to fully exploit the benefits of heterogeneous hardware.
The Motivation

- **Innovative** architectures, algorithms, and specialized programming environments and tools are needed to efficiently use these heterogeneous hardware.

- **Approach**
  - Use a top-down approach to propose a reference architecture
  - Consider requirements engineering, software design, parallel programming environments, and heterogeneous distributed/parallel architectures
  - Optimisation should include
    - energy efficiency
    - performance
    - Other factors: dependability, data movement, security and cost-effectiveness
Keep this to the min, Aim for the max!

Many diverse Applications

Define a Common Architecture

Catalogue of Tools and Technologies

what's the best path?

Heterogeneous hardware
TANGO Architecture

**IDE:** modelling, design and construction of applications

**Middleware:** application deployment and placement considering energy/performance requirements on target heterogeneous parallel architectures.

**Fabric:** Heterogeneous parallel devices management
Open and freely accesible set of tools for managing heterogeneity

Downloadable from: http://www.tango-project.eu/content/final-tango-toolbox-released-open-source

Two usage workflows are defined as a matter of example
TANGO tools (I)

**Placer** is a design-time tool that thoroughly optimizes the placement and scheduling of complex software onto heterogeneous multi processing hardware platforms.

**DTC-Poroto** enables the generation and early performance characterisation of FPGA-offloaded kernels for user defined computations.

**Code Profiler** is a tool for analysing Java code for its energy efficiency.

**Programming Model and Runtime Abstraction Layer** is a combination of COMPSs and OmpSs task-based programming models, which simplifies the development of parallel application for distributed heterogeneous platforms.
Monitoring Infrastructure monitors the heterogeneous resources to provide metrics about the status of the various devices and also historical statistics.

**ALDE** is responsible for the workload scheduling and the management of the application life-cycle while it is executed.

**Self-Adaptation Manager** manages the runtime-based adaptation strategy applied to applications and Heterogeneous Parallel Devices.

**Energy Modeller** forecasts future application and host power consumption, as well as reporting current and historic energy usage.
Device Emulator finds an efficient mapping of the application tasks onto the nodes/cores in low time, i.e., which application task should run on each node/core.

Device Supervisor is an extension of Slurm (JobPack) which is an open-source cluster resource management and job scheduling system.

CrypTANGO is the security solution for TANGO. Using the CrypTANGO libraries, one can develop TANGO applications which securely compute over sensitive data.
TANGO for HPC

HPC Developer

TANGO TOOLBOX

- Programming Model
- App src
- ALDE
- App Binaries
- IDE
- Self-Adaptation Manager
- Energy Modeler
- Device Supervisor
- Monitor Infrastructure
- App Scheduling
- Heterogeneous Devices
TANGO for embedded
TANGO in Use

Source code of an application available, parallel matrix multiplication.
Same application without modification to be executed:

1. CPU
2. GPU
3. CPU + GPU
ALDE: automatic deployment

Source code from Programming Model

Creates a Singularity Container

Uploads Container to Device Supervisor

Execution Configuration 1: CPUs

Execution Configuration 2: GPUs

Execution Configuration 3: CPUs + GPUs
It is the software responsible for the resource management and allocation of jobs.

- It matches the jobs needs for compute power to the resources characteristics
- It maintains queues of jobs and provides scheduling based on particular constraints and algorithms

- It can supervise several nodes that contain heterogeneous devices
- It follows the job execution and can provide accounting and profiling results once the job is finished
- It offers various APIs that can be used by the ALDE.
- It is represented by the open-source RJMS Slurm.
Monitoring infrastructure

- It monitors the heterogeneous resources
- Specially in terms of energy usage, utilization and temperature
- It relies on Collectd and a set of custom plugins (for NVIDIA GPUs and Xeon Phi coprocessors)
- It uses InfluxDB (historic values) and Grafana (GUI)
- It offers a REST API to connect to other TANGO components
  - Self Adaptation Manager, Energy Modeller, PM
TANGO in Use

CPU execution

Time: 61.195 ms
Energy: 13.10 Kilo Joules
TANGO in Use

GPU execution

- Time: 31.041ms
- Energy: 8,28 Kilo Joules

HPC Developer

TANGO TOOLBOX

Programming Model

ALDE

App Binaries

IDE

App src

Self-Adaptation Manager

Device Syst

Energy Modeler

App Scheduling

Monitor Infrastructure

Heterogeneous Devices

GPU execution

GPU execution
TANGO in Use

- **CPU + GPU execution**
- **GPU + CPU execution**
  - Time: 30.685 ms
  - Energy: 8.43 Kilo Joules

**TANGO TOOLBOX**
- HPC Developer
- IDE
- Programming Model
- App src
- ALDE
- Device Scheduling
- Self-Adaptation Manager
- Energy Modeler
- Monitor Infrastructure
- Heterogeneous Devices
- CPU + GPU execution
## Measurements

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Time</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong> 12 cores, 2 nodes</td>
<td>61.195 ms</td>
<td>13.10 Kilo Joules</td>
</tr>
<tr>
<td><strong>GPU</strong> 2 nodes</td>
<td>31.041 ms</td>
<td>8.28 Kilo Joules</td>
</tr>
<tr>
<td><strong>CPU + GPU</strong> GPU node + 12 cores CPU</td>
<td><strong>30.685 ms</strong></td>
<td>8.43 Kilo Joules</td>
</tr>
</tbody>
</table>
## Users - Components Mapping

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
<th>Components</th>
</tr>
</thead>
</table>
| Software/Hardware Designer | • Combine energy-awareness with the principles of requirements engineering and design  
                              • Specify performance/energy goals to evaluate the efficiency of an application | • Placer  
                              • Design Time Characteriser  
                              • Device Emulator |
| Software Developer    | • Program applications on heterogeneous distributed hardware  
                              • Produce energy efficient code | • Code Profiler  
                              • Programming Model |
| App. Deployer - Business | • Measure the energy consumption / map the energy profile for the application  
                              • Identify the workload profile and its energy model  
                              • Provide actual energy measurements as KPIs | • Energy Modeller  
                              • Self-Adaptation Manager  
                              • Device Emulator |
| App. Deployer - Technical | • Deploy application/services in an energy efficient environment | • ALDE  
                              • Device Emulator |
| HW Infrastructure Manager | • Use energy placement and power management policies  
                              • Schedule/manage hardware devices efficiently  
                              • Monitor energy consumption and adapt accordingly | • Device Supervisor  
                              • Monitoring Infrastructure  
                              • Self-Adaptation Manager |
| Integrator            | • All of the above | • All components |
Validation of toolbox

- Remote processing
- Embedded water meters
- HPC miniapps (Gromacs and Cloverleaf)
Impact on real market

- **Major impact in embedded world:**
  - Programming Model can be a breaking advance in parallel programming for embedded applications.
  - Placer helps to identify the best HW devices and platforms during the design phase

- **Major impact in HPC world:**
  - Energy Modeler can define new way to understand and manage the applications
  - Annotation of code and automatic deployment avoids re-compilations of code when HW platform changes
Get connected...

- Visit our web site: [www.tango-project.eu](http://www.tango-project.eu)
- Follow us on Twitter @TANGOModel
- Subscribe to our Newsletter
- Download and contribute to our tools
- Or contact us by email: clara.pezuela@atos.net
Thank you!