Alliance on the Road: Architecture, Catalogue and Tools

Karim Djemame
School of Computing
University of Leeds, UK
pursuing a common objective with the H-HW&SW Alliance: influence and develop the heterogeneous market.
Outline

- Alliance: Motivation
- Towards an architecture
- Catalogue and tools
- Conclusion
Context

Need to design more flexible software abstractions and improved system architectures to fully exploit the benefits of heterogeneous hardware.
Traditional programming approaches for parallel algorithms, programming environments and tools at best achieve a small fraction of the efficiency and the potential performance that we should expect from parallel computing in computing systems which are:

- Highly diversified
- Operate in mixed environments
- Based on heterogeneous architectures.

Grids

Clouds
Heterogeneity has emerged as one of the most profound and challenging characteristics of these parallel environments.

Two levels are identified:

- **Macro** level: networks of distributed computers (clouds, Grids, clusters), composed by diverse node architectures (single, multi-core), are interconnected with potentially heterogeneous networks.

- **Micro** level: deeper memory hierarchies (main, cache, disk storage, tertiary storage) and various accelerator architectures (fixed, programmable, e.g. GPUs, and reconfigurable, e.g. FPGAs).
Heterogeneity ... Causes

1. Multiple types of programmable core
   - CPU (lightweight, heavyweight)
   - GPU
   - Others (accelerators, ...)
2. Software (OS, middleware, tools, ...)
3. Interconnect asymmetry
4. Memory hierarchies

- Note: Heterogeneous System Architecture (HSA) is designed to efficiently support a wide assortment of data parallel and task-parallel programming models
The impact of heterogeneity on all computing tasks is rapidly increasing.

Innovative architectures, algorithms, and specialized programming environments and tools are needed to efficiently use these new and mixed/diversified parallel architectures.

Idea:
- Use a top-down approach to propose a reference architecture
- Consider requirements engineering, software design, parallel programming environments, and heterogeneous distributed/parallel architectures.
The Proposed Approach

- Use a (holistic?) approach to hide the complexity between the heterogeneous hardware level and the level of application/software
- Identify **missing functionalities** to support key quality factors across the reference architecture
- Define and integrate **measures** of key quality factors into the design and development process for software running on heterogeneous hardware
- Evaluate the **impact** of patterns onto key quality factors

- Consider **trade-offs** in terms of:
  1. Increased software complexity
  2. Increased programming burden
  3. Increased architectures heterogeneity
Heterogeneity: Value

Keep this to the min, Aim for the max!

Define a Common Architecture

Catalogue of Tools and Technologies

Many diverse Applications

what’s the best path?

Heterogeneous hardware
Layered Architecture?

Applications

Programming Models

QoS Negotiation, Admission Control, Execution Management, Monitoring, Metering, Accounting

Apps Hosting Platforms

Core Middleware

User-Level Middleware

User level

System level

HPA Resources
Application Life Cycle

Many diverse Applications

Application design

Application construction

Application deployment

Application operation

Heterogeneous hardware
Layered Architecture and Factors

- QoS Negotiation
- Admission Control
- Execution Management
- Monitoring
- Metering
- Accounting

Programming Models

Applications

Stack

Configuration

Management

Core

Middleware

User level

User-Level Middleware

Core Middleware

System level

HPA Resources

Performance

Time-Criticality

Data Movement

Security

Cost-effectiveness

Self-Management

Stack Configuration Management
Alliance - Progress
(June - December 2017)
Alliance - Towards an Architecture

- “Architecture” layers and dimensions
- How/what components are connected together
- What HW architectures to target
- Possible infrastructure and kind of communication to support
- Other
Approach

- Taxonomy of applications
  - Projects: TANGO, HERCULES, P-SOCRATES, RAPID, SHARCS, ECOSCALE
  - Other projects (Vineyard, E2DATA, OPERA)
- Elicitation of application requirements
- Drive future steps following this exercise
  - Architecture, programming models, platforms, energy, QoS, real-time, security, virtualisation ...
What Do We Need to Investigate - Themes

- Applications
- Platforms
- Programming Models
- Programming - Tools Support
- Middleware
- Virtualisation
- Quality of Service
- Low Power/Energy Computing
- Predictability
- Security (including trust and privacy)
- Secure Updates
- Other: disruptive applications, disruptive technologies (e.g. edge)

Think catalogue and tools
Applications

- High Performance Computing
- Big Data
- Cloud computing
- Internet of Things
- Embedded systems
- Real-Time systems
- Automotive and avionics
Key Factors Addressed

1. Performance
2. Energy Efficiency
3. Cost
4. Dependability
5. Security
6. Real-Time
7. Other (Programmability)
Hardware Heterogeneity

- ManyCore CPU (as host system)
- GPGPU (as host system)
- FPGA (as accelerator)
- SoC (as accelerator)
• To include the preliminary findings regarding the applications (general) requirements in the working document
  • Draft last version November 2017
• Questionnaires - technical
• Establish working groups:
  • Focus
  • Mission and specialisation
  • Reporting
Conclusion

- Range of applications continue to grow
- Presented Alliance aim and research approach from the technical perspective
- Need for a reference architecture to support and benefit from heterogeneous hardware
  - Strong emphasis on:
    - Application life cycle
    - Key aspects: performance, energy efficiency, time-criticality, dependability, data movement, security, cost-effectiveness
- Catalogue and tools to support of application construction, deployment, and operation
Thank you!