The BarbequeRTRM[2]

A run-time resource manager for multi/many-core architectures

Patrick Bellasi, Giuseppe Massari and William Fornaciari
Dipartimento di Elettronica Informazione e BioIngegneria, Politecnico di Milano, Italy

Problem

Modern and future many-core platforms are going to offer considerable computational capabilities. Current parallel programming models (e.g. OpenCL, OpenMP, ...) aim to exploit such architectures focusing on the performance maximization. Mobile and embedded systems needs applications that could adapt their level of parallelism according to changes in resource availability, due to contention, power budgets and response to critical thermal conditions. Therefore, efficient run-time resource management (RTRM) techniques [1] are a key goal in the usage of these platforms. The BarbequeRTRM framework proposes a solutions that overcome the lacks of general-purpose operating systems, performing a multi-objective resource partitioning and supporting reconfigurability of run-time managed applications.

Run-Time Resource Management

RTRM covered requirements and provided support

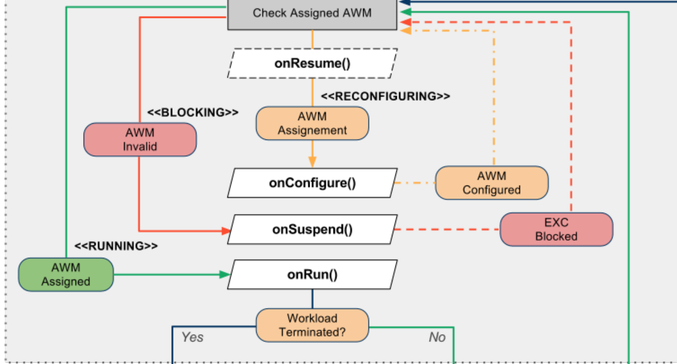
Requirements	Description
monitor resource status	Export an updated view of the **usage and behaviors of each subsystem**, and their resources, with different details levels.
dynamic resource partitioning	**Grant resources** to critical workloads while dynamically yield these resources to best-effort workloads when (and only while) they are not required by critical ones, thus optimize resource usage and fairness.
resource abstraction	Handle a **decoupled resource view** between the users and the underlying hardware.
multi-objective optimization policy	**System-wide** and multiple metrics **optimization policy**.
dynamic optimization policy	Support **multiple and tuneble policies** to fit well specific optimization scenarios.
low run-time overheads	**Reduced impact** on the performances of the controlled system.

The BarbequeRTRM is a **framework for adaptive RTRM of many-core architectures**, offering optimal **resource partitioning** and **adaptive run-time scheduling** of the different **reconfigurable applications**. It has been designed to be the core of an highly modular and extensible run-time resource manager, providing support for an easy integration and management of multiple applications competing on the usage of one (or more) shared MIMD many-core computation devices. It is possible to support both homogeneous and heterogeneous architectures, since the the low-level communication (*Platform Proxy)* is handled by a system specific module.

Management policies optimize resource assignment to demanding applications considering:

- *application properties*, e.g., resource requirements, Quality-of-Service and relative priorities
- *resources availability and state*, e.g. frequency, power consumption, process variation and thermal conditions.
- *tunable run-time optimization goals*, e.g., power reduction, energy optimization, reconfiguration overheads minimization, congestion avoidance, fairness and overall performances maximization.

The BarbequeRTRM framework

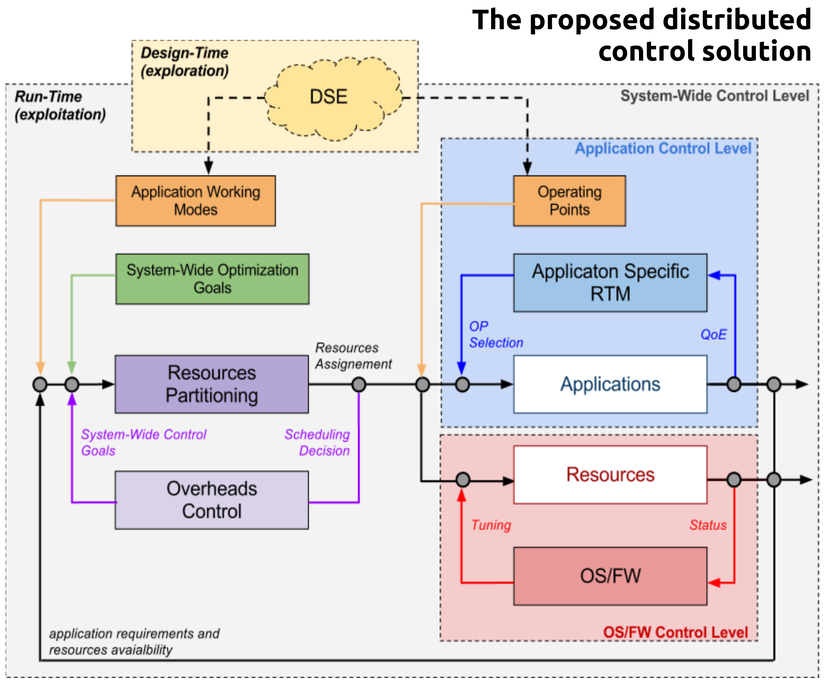


At **design time**, a suitable *Design Space Exploration (DSE)* activity identifies a set of resource requirements that are worth to be considered at runtime, namely *Application Working Modes (AWM),* and a set of application specific parameters defining different QoS levels, namely *Operating Points (OP*).

At **run-time**, the resource management is enforced in a **hierarchical and distributed** way:

- For each running application, the RTRM assigns the most promising AWM, as a result of a **system-wide multi-objective optimization**.
- The application can optionally perform a **QoS fine tuning** by switching among its OPs.
- Platform specific mechanism (e.g., DVFS) are exploited to avoid risky conditions.

The proposed distributed control solution



Application integration and Results

Abstract Execution Model

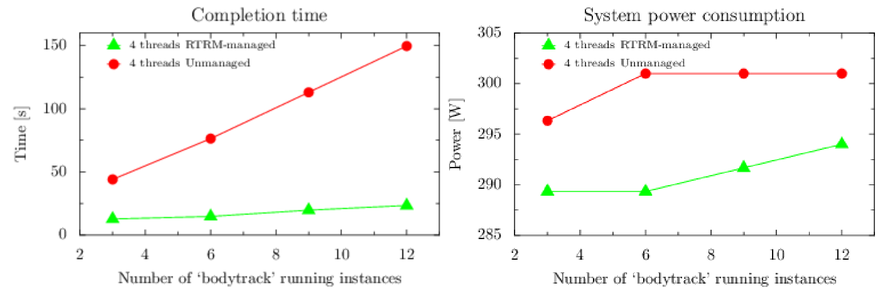
The framework provides a library (*RTLib*) for the integration of the application with the RTRM. In the RTLib a base C++ class is defined (*BbqueEXC)* exporting a small set of **callback methods**.

From the RTRM point of view, each independent module of the application represents an *Execution Context (EXC)*, and must be implemented as a class derived from BbqueEXC. Therefore the RTRM manages the execution flow of the EXCs, by invoking the methods according to the *Abstract Execution Model (AEM)*.

The AEM allows the EXC to be aware of changes of resource availability. Considering a typical parallel application, this means to adapt its level of parallelism, i.e., number of running threads, or OpenCL kernels; along with redefine usage of memory by resizing the data buffers, and so on.

Initial Power Optimization Experiments

Several instances of *bodytrack* concurrently running on 3 quad-core NUMA nodes



- In the *unmanaged* case the resource assignment is charge of the Linux scheduler, while in the *RTRM-managed* is the scheduling policy of the BarbequeRTRM that defines resource partitioning by properly configuring CGroups at run-time.
- A considerable improvement in the completion time has been reported, due to the better capability of the RTRM to manage concurrency by enforcing resources partitioning.
- In highly congested scenarios, the optimal partitioning and an eventual sequential scheduling, reduces access conflicts and corresponding effects (e.g. caches trashing, time delays).
- Lower power consumption, and greater energy benefits also, since the completion time is reduced.

References

[1] A. Bartzas et al. *"Run-Time Resource Management Techniques for Many-Core Architectures: the 2PARMA Approach"*, ERSA'11 International Conference on Engineering of Reconfigurable Systems and Algorithms, Las Vegas, USA, July 18-21, 2011.
[2] P.Bellasi, G.Massari, W.Fornaciari, *"A RTRM proposal for Multi/Many-Core platforms and reconfigurable applications"*, ReCoSoC'2012 IEEE International Workshop on Reconfigurable Communication-centric Systems-on-Chip, York, UK, July 9-11, 2012.
[3] P.Bellasi, G.Massari, W.Fornaciari *"System-Wide run-time resource management for multi-many cores in the 2PARMA Project"*, Session on Power-Efficiency and Program Correctness Analysis for Scalable Multicores, HiPEAC Computing Systems Week April 2012, Göteborg, Sweden, April 24-25, 2012.

http://bosp.dei.polimi.it

SEVENTH FRAMEWORK PROGRAMME

2 PARMA http://www.2parma.eu

{patrick.bellasi, giuseppe.massari, william.fornaciari}@polimi.it