Run-Time Resource and Memory Management in MANGO platform

Federico Reghenzani*[1],
Anna Pupykina*[1], Giuseppe Massari*[1],
Giovanni Agosta*[1],
William Fornaciari*[1]

* *DEIB Politecnico di Milano, via Ponzio 34/5, 20133 Milano, Italy*

ABSTRACT

Current and future High Performance Computing systems come not only with increasing computational capabilities, but also with a set of non-functional requirements to meet. Moreover, real-time predictability can represent a further requirement for some systems. In this scenario, especially when characterized by complex and heterogeneous architectures, a middleware for the resource management becomes essential. In this work, we briefly discuss the implementation of the resource manager at node-level in MANGO project, that involves complex hardware and software architectures, due to the deeply heterogeneous configuration of the proposed computing infrastructure. We present our partition-based management approach, with a focus on the prediction-based memory management adopted.

KEYWORDS: High Performance Computing; Resource Management; Memory Management; Heterogeneous Architectures

1 Introduction

High Performance Computing is quickly evolving at the hardware, software and application levels: (1) heterogeneity emerges as a dominant trend for pure performance and performance per watt; (2) new classes of applications emerge; (3) a push towards cloud HPC [KG15] aimed at providing computational resources to classes of users which could not afford them in the past. In this context, time-critical applications, such as financial analytics, online video transcoding, and medical imaging require predictable performance, which are at odds with the need of maximizing resource usage while minimizing power consumption.

Extending the traditional optimization space, the MANGO project [FAA+16] aims at addressing what we call the PPP space: *power*, *performance*, and *predictability*. The MANGO European project investigates the architectural implications of the emerging requirements of HPC applications, to then define a new generation high-performance, power-efficient, deeply heterogeneous architectures with native mechanisms for isolation and QoS compliance. The MANGO HPC infrastructure is a distributed system, featuring computational

[1]E-mail: {name}.{surname}@polimi.it

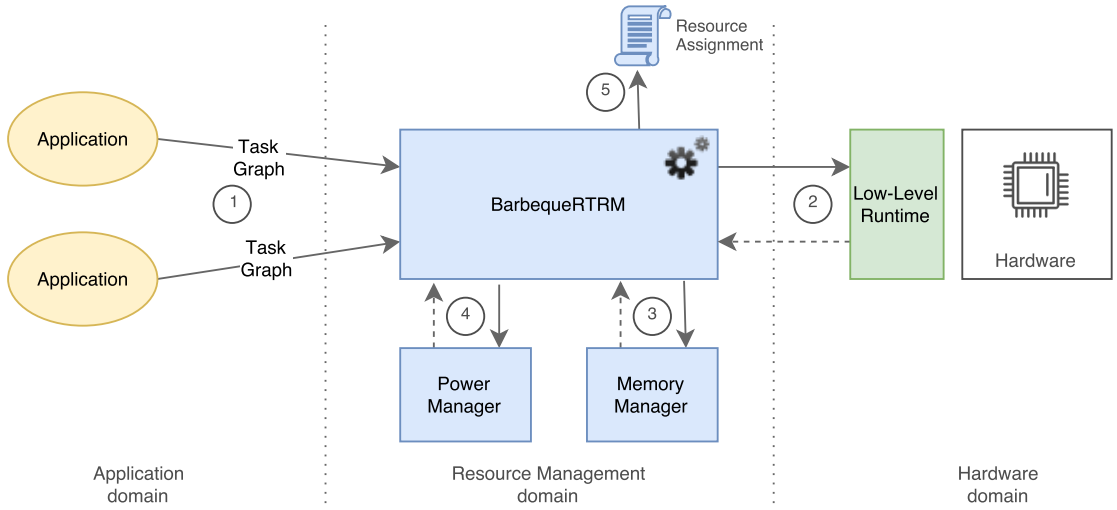

Figure 1: The MANGO partition-based resource management: (1) the applications provide their task graphs to BarbequeRTRM that (2) requests available partitions to the low-level runtime, then (3) (4) it asks to memory and power managers to skim the partition list and to assign a score to each partition. Finally, it selects the best partition and enforces (5) the resource assignment.

nodes composed of general-purpose processors (GN), linked via PCIe or Ethernet with a set of heterogeneous accelerators (HN) interconnected through a NoC. This scenario requires the management and allocation of resources among different applications in a way that maximizes resource usage, while preserving the predictable execution time of critical applications, and the expected power budget.

2 A partition-based resource management

The approach to resource management in the MANGO project is a partition-based schema, in which the **Barbeque Run-Time Resource Manager** [BMF12] has only partial authority in the resource assignment decision process. This is primarily due to the lack of full observability of hardware resources in the system. However, even considering a full observable hardware, developing a monolithic decision algorithm for the huge decision space would be too complex, as subsequently analyzed. It would consequently lead to an inefficient and/or ineffective resource assignments.

The applications have to expose to the resource manager a **task graph** describing its structure, in terms of kernels and buffers to be allocated to the heterogeneous node.

In the MANGO project, selecting the best allocation for the task graph depends on several factors: (1) the available processor types and the target architectures of the application kernels; (2) the number of threads of the kernels and the number of cores of the processors; (3) the NoC resources, in terms of bandwidth and availability of virtual networks; (4) the memory allocation, especially considering the internal fragmentation; (5) the temperature and power requirements at different layers, from the single core to the entire infrastructure; (6) the potential cache conflicts and optimizations; (7) the application QoS and its variability.

Taking in account all of these factors entails a multi-dimensional and large decision space, especially in the presence of multiple applications. The complexity of finding the

optimal solution would be not feasible for the computational power available today.

In order to efficiently explore this decision space our resource manager uses a partition-based schema, depicted in Figure 1. A **partition** is a subset of available resources that can be assigned to a **task graph**, able to guarantee the application requirements. The list of available partitions is provided by several *actors* that are able to add or remove partitions from the list. In the case of MANGO project, the actors are: (1) the underlying hardware run-time that provides the list of feasible partitions in terms of NoC resources, (2) the memory manager (described in Section 3) that removes the unfeasible partitions in term of memory availability and fragmentation, (3) the power manager that removes the unfeasible partitions in term of temperature and power control. These actors reduce the decision space of the resource manager, allowing the development of high-level policies in BarbequeRTRM.

Besides the removal of unfeasible partitions, each actor should provide a *score* in the range $[0; 100]$ for each partition. How this score is calculated is delegated to an own policy of each actor. Instead, the high-level policy of the resource manager is in charge of selecting the best partition according to the scores provided by each actor and according to other system metrics. Consequently, the decision process is distributed among the several policies of each actor that contribute to the final decision of the resource manager. How to select policy algorithms and how to tune them are critical tasks in the design of the resource manager.

3 Prediction-based Memory Management

The MANGO architecture is based on a shared memory among all the heterogeneous units in a node. To efficiently manage the available memory resources, we design a memory manager that serves memory requests in a resource allocation-aware fashion, employing knowledge about the evolution of the workload to maximize the utilization of resources while optimizing the ability of the node to serve high priority applications.

The memory management system deals with the following objectives: (1) choosing the most suitable memory according to the allocated processing elements; (2) enabling concurrent, thread-safe memory allocation and deallocation while avoiding fragmentation; (3) performing translation from virtual to physical addresses and vice versa; (4) performing runtime optimization.

Challenge is understanding the best memory unit for a given kernel or group of kernels. We are given a buffer of size S, a unit or set of processing units U which uses this buffer and a partition or set of partitions $P = \{M, U\}$ appropriate to allocate buffer, where M is a memory unit. We evaluate M from P to select the best solution for future memory usage. For future memory usage it is necessary: (1) to allocate the buffer in memory unit near processing unit and leave memory units free near unused processing units; (2) to leave free spaces for allocation of high priority requests.

To achieve the first goal we propose to use a fuzzy multi-criteria analysis with pairwise comparison. The multi-criteria analysis is a decision-making tool developed for complex multi-criteria problem. It should be noted that it is more difficult to estimate the significance of some particular criteria than to determine the best of the two. Pairwise comparison is used to avoid direct assignment of weights or scores criteria to the available options.

To maximise the ability to allocate future high priority requests, the choice between memory units will be based on a statistical prediction of the future memory state. In general, a memory management system is based on an algorithm that takes runtime decisions on the

basis of continuously updated information about the state of the resources. By predicting the future state of resources, in these cases, we can improve the quality of the management decisions.

We implemented two linear prediction models that can be applied to runtime contexts (Moving average and Exponential weighted average methods) and considered two variants to form statistical series:

- Based on time - to obtain the series, it is necessary to carry out a sampling. Since the timing of allocation and deallocation actions for different memory units may not be aligned with the sampling, then the size of the sampling step may adversely affect the prediction. In the current version of the algorithm we use a constant sampling step, although it could be possible to use an adaptive algorithm for more accurate results.
- Based on event - consider each allocation as the occurrence of an event. In view of the fact that events have variable frequencies, events may have less or more impact on prediction than needed.

We investigated the developed algorithms through a simulation-based approach. The algorithm with prediction based on moving average method achieves the highest success rate, both overall and when considering only high-priority requests. In general, all versions of the prediction-based algorithm are able to serve a higher ratio of high priority requests with respect to the baseline. Through the use of predictive algorithms, we are able to serve up to 82% of the high priority requests vs a baseline of 54% without prediction.

4 Conclusions

In this work we presented our partition-based resource management schema currently implemented in the MANGO project, with a description and preliminary results of the prediction-based memory management. This approach simplifies the decision process distributing the policy among several actors. Even if the resource allocation is in general sub-optimal, this partition-based resource management enables the possibility to deal with heterogeneous requirements of current and future HPC systems.

References

[BMF12] Patrick Bellasi, Giuseppe Massari, and William Fornaciari. A rtrm proposal for multi/many-core platforms and reconfigurable applications. In *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2012 7th International Workshop on*, pages 1–8. IEEE, 2012.

[FAA+16] José Flich, Giovanni Agosta, Philipp Ampletzer, David Atienza Alonso, Carlo Brandolese, Alessandro Cilardo, William Fornaciari, Ynse Hoornenborg, Mario Kovač, Bruno Maitre, et al. Enabling hpc for qos-sensitive applications: the mango approach. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*, pages 702–707. IEEE, 2016.

[KG15] Bastian Koller and Michael Gienger. Enhancing high performance computing with cloud concepts and technologies. In *Sustained Simulation Performance 2014*, pages 47–56. Springer, 2015.