D3.9 Final Report on Probabilistic Analysis for Mixed Criticality on Manycore Version 1.0

Document Information

Contract Number	611085
Project Website	www.proxima-project.eu
Contractual Deadline	m25, November-2015
Dissemination Level	PU
Nature	R
Authors	Iain Bate (UoY), David Griffin (UoY), Benjamin Lesage (UoY), Frank Soboczenski (UoY), Jaume Abella (BSC), and Enrico Mezzetti (UPD)
Contributors	BSC, UPD, RPT, UoY
Reviewer	UPD
Keywords	Measurement-Based Probabilistic Timing Analysis, Manycore, Time-composable

Notices:
The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n^o 611085.

Change Log

Version	Description of change
v0.1	Initial Draft for internal review
v0.2	Internal review comments addressed
v1.0	Initial version released to the European Commision

Contents

Executive Summary

The present document is part of the final and third milestone of the PROXIMA project. This deliverable focuses on the analysis of inter-core interferences, that is how effects external to the analysed task, i.e. co-runners on the different cores, can impact its temporal behaviour. We present a qualitative evaluation of the PROXIMA approach to mixed-criticality systems (MCS) where tasks of different criticality execute concurrently; the intra-core timing analysis and the derivation of inter-core interference models are themselves presented in extensive details in D3.8.

The communication infrastructure and shared resources play a central role in the extent to which inter-core interferences manifest themselves [7,10,11]. The PROXIMA approach to MCS, through both hardware and software mechanisms, aim to provide timing composability and performance isolation. Composability reduces the dependence between observed inter-core contention and execution time caused by shared resources. Performance isolation caters to the highest criticality tasks, without the cost of physically partitioned platforms or over-provisioning incurred by multi processors.

The Bespoke Manycore Platform (BMP) and its alternatives have been designed to be MBPTA-compliant (Measurement-Based Probabilistic Timing Analysis) while providing the desired level of time-composability. They further adhere to the principles of composability. The BMP and the PROXIMA approach to timing analysis for manycore have been presented in MS2 D3.5.

The BMP has been further extended to explore alternative designs, in particular topologies prevalent in COTS HW manycore platforms. This document is a qualitative assessment as to how those alternatives, adapted to follow the PROXIMA philosophy, impact the requirements behind the timing analysis techniques detailed in D3.8, and the challenges involved in meeting those requirements. In particular, we discuss how timing estimates obtained for a task on the extended BMP, as derived using the techniques detailed in D3.8, remain valid irrespective of the contention experienced on the final system.

Difficulties related to the integration of the OS and the simulator code bases prevent the evaluation of the complete manycore platform. A relevant discussion is included in D2.10. The quantitative evaluation of the benefits of the hardware platform, as compared to prevalent COTS designs, is available instead in D1.12. We only include preliminary results pertaining to the effects of interferences in non-MBPTA compliant, deterministic manycore designs.

1 Introduction

This deliverable discusses the extent to which Measurement-Based Probabilistic Timine Analysis (MBPTA) can be applied to the Bespoke Manycore Platform (BMP) developped in the PROXIMA project. As a qualitative assessment, it presents how the philosophy behind alternative designs of the BMP contributes to the time-composability of the derived estimates. We further focus on the impact of alternative design choices, over the baseline PROXIMA manycore architecture, as they prevail within COTS HW platforms. A similar discussion regarding the impact of the baseline PROXIMA manycore architecture can be found in D3.5 (MS2). Hardware and software techniques are combined to reduce the dependencies between the observed execution time and inter-core contention. The focus on composability within PROXIMA further lessens the dependencies between concurrent tasks of different criticalities, as may occur within mixed-criticality systems (MCS). This deliverable therefore introduces the following:

1. *Platforms* - the Bespoke Manycore Platform (BMP) used for the evaluation of alternative architectures is first introduced. The platform, both architecture and supporting operating system (OS), is briefly described in Section 2 of this report. The underlying architecture and OS are detailed in their respective deliverables, D1.12, and D2.10.

2. *Timing Analyses* - The impact of the proposed manycore platform on MBPTA (D3.8) is discussed in Section 3. Section 4 discusses the application of the inter-core interference analysis presentsed in D3.8 to the BMP.

The contents of Section 2 are largely covered in the MS2 deliverables D1.12 and D2.10. We only provide a brief summary of those documents. Section 3 is the centre of the present document. Please note that in order to be self-sufficient, the present document builds upon D3.5, taking into account modifications and additional results regarding the underlying platforms. Section 3 focuses on a qualitative evaluation of the impact of the platform because of difficulties in the integration of the BMP, as discussed in D2.10. The application of the PROXIMA timing analysis to the platform is evaluated in D1.12.

2 Overview of the Platform and Simulator

The manycore platform is developed in the context of WP1 – first described in D1.2 and D1.5 (MS1), and then revised in D1.8 (MS2) and D1.12 (MS3). For the sake of brevity, this section only provides an overview of the manycore platform, operating system and simulator. An extensive description of both the platform and the simulator is located in D1.8 (MS2). D1.12 (MS3) introduces and evaluates design alternatives allowing more refined communication channels. While outside the scope of the PROXIMA manycore platform, presented in D3.5, we discuss their potential in terms of analysis.
The operating system, ManyCOS, is developed in the context of WP2 and a more complete description is available in D2.7 and D2.10. This section also describes the tracing capabilities of the simulator, vital to the definition of timing analysis methods. Design choices from the perspective of the timing analyses are discussed in Section 3.

2.1 *Architecture of the Manycore Platform*

The simulator models a clustered manycore platform. Each cluster typically contains a local memory and 4 to 16 cores, although a different number of cores are possible. Each core features local first level data (DL1) and instruction (IL1) caches, with respective translation look-aside buffers (DTLB and ITLB). A L2 cache further supplements each cluster's local memory hierarchy. The L2 cache is shared among all cores in the cluster through an intra-cluster tree-based Network-on-Chip (NoC). Hence, the infrastructure of a cluster is akin to that of a multicore platform using a tree-based interconnect between the cores and the main memory system (including the L2).

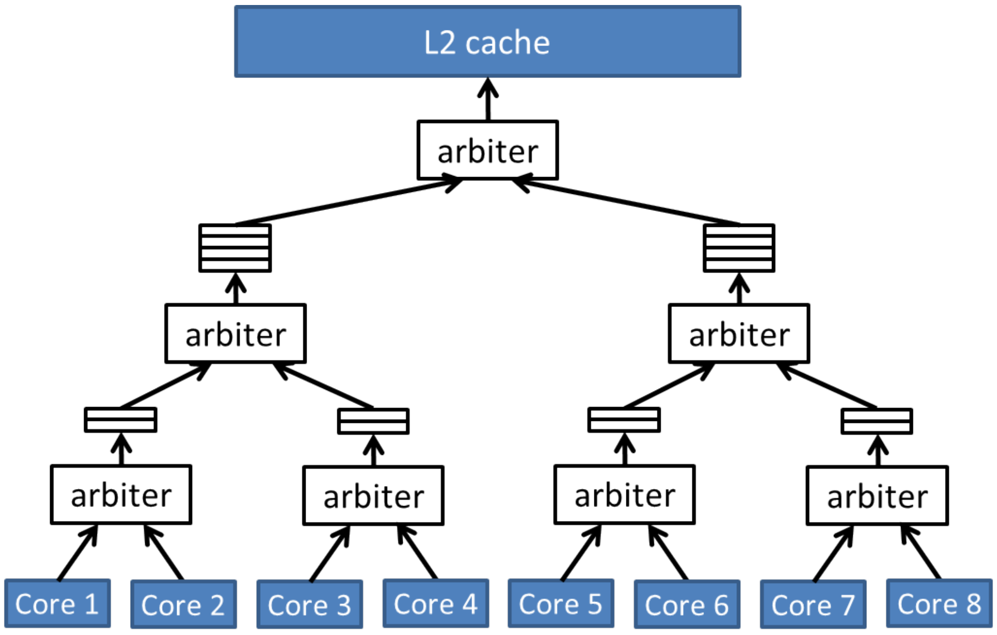


Figure 1: Example of an 8-core cluster manycore connected with an intra-cluster tree NoCs.

A number of MBPTA-compliant policies have been implemented for the different layers of the local memory hierarchies. Caches, private or shared, and TLB use randomised placement and replacement policies. The shared L2 cache further supports partitions, exclusive to specific cores or shared among all of them. D1.8 demonstrates the superiority, for the NoC, of policies based on random permutations of the arbitration schedule [5], as opposed to bounded round-robin [8] or purely random (lottery) policies. Each L2 cache is connected to a memory controller which enforce the maximum arbitration latency during timing analyses, upper-bounding its deployed behaviour [8].
The intra-cluster tree NoC supports heterogeneous bandwidth requests. Each arbiter can be configured to grant more frequent access to one of its input ports. The configurable allocation of bandwidth allow for tighter timing estimates for the highest priority cores. Heterogeneous bandwidth provides a trade-off between the tightness of timing estimates and the guarantees available to all cores.
To allow for N-N communication between clusters, the inter-cluster infrastructure has been modified. Clusters are interconnected through a wormhole-based, meshed NoC to allow for cluster-to-cluster communication from one local memory to the other. To each cluster is attached a router, connected to its four neighbours: North, East, West and South. The router directs packets, divided in flits, from its inputs to either of its outputs, including its attached cluster. Outputs buffer are dedicated to a packet until its tail flit has been transmitted. Routing is performed using the XY algorithm: packets first travel East or West to align with their destination, then North or South.
The arbitration of output ports in each router relies on a randomised policy. A A new arbitration window, a permutation of the input ports requesting access to output, is randomly generated once a windows has been used. It is a work-conserving algorithm as the arbitration pointer skips over inputs with no pending requests. Worst-case contention can be enforced by injecting requests at the maximum possible rate for each contending node on the analysed flow route.
Overall, local memory requests, having to traverse only the intra-cluster NoC, are served much faster than remote ones. Requests to distant memories must, in order, go through (i) the intra-cluster NoC of the emitting cluster, (ii) the inter- and (iii) intra-cluster NoCs of the receiving cluster.

2.2 Tracing Capabilities

The manycore simulator interfaces with Rapita Verification Suite (RVS) to produce traces in the appropriate format; the simulator captures instrumentation points in the application and reports the cycles in which they have been observed. Neither the instrumentation nor the tracing interferes with the timing simulation. The binary is unchanged whether instrumentation is active or not, while tracing occurs at emulation level. RVS support is provided without affecting the timing of the simulated application.
The tracing is highly flexible and the numerous end-of-simulation statistics could be reported at a finer granularity, such as that of the iPoint. The availability of such information is of prime importance in the application and evaluation of MBPTA methods suited to the BMP. A non-exhaustive list of those statistics includes execution cycles, hits and misses in each level of the memory hierarchy,

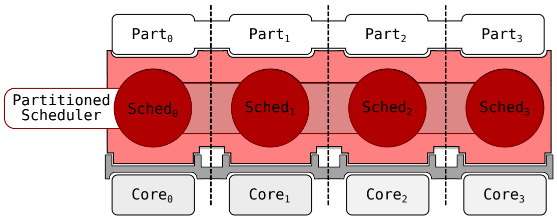
accesses to the NoC, stall cycles due to contention, etc..
Therefore, the simulator enables many ways of collecting a wide spectrum of information so that manycore-specific timing analyses, like the interference analysis proposed in D3.8, can be effectively investigated and released. The malleability of the simulator also allows for the exploration of alternate sources information, in line with the instrumentation facilities provided by concrete hardware platforms, for the proposed analyses.

2.3 Architecture of the operating system ManyCOS

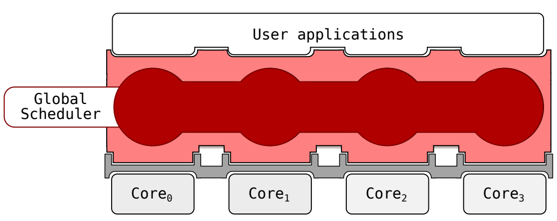
A custom RTOS, first described in D2.7, is under development to support the execution of mixed-criticality applications on top of the PROXIMA manycore simulator. Alternatives at the platform level do no impact the RTOS design philosphy, the following description matches the one in D3.5 and is included for self-sufficience. The current status of the RTOS integration is discussed in D2.10. The definition of a PTA-compliant hardware architecture releases the software layer and RTOS from the need to guarantee that the whole system is amenable to PTA. The features implemented within the manycore RTOS aim at guaranteeing a better applicability of timing analyses in general and are intended to be equally convenient and useful in probabilistic and deterministic platforms.
The architectural design of the RTOS has been steered by two main principles: (i) support to mixed-criticality systems and (ii) time composability, as a fundamental enabler to incremental development and qualification. A mixed-criticality RTOS is required to support the execution of applications or functions characterized by different levels of criticality, as defined by the specific certification standards applied in the domain.
Hypervisor-based solutions do not scale smoothly to manycore platforms where support to mixed-criticality applications is provided with a combination of hardware partitioning (e.g., resorting to a clustered architecture) and bounded resource usage. With this respect the RTOS is expected to exploit the functionalities of the underlying hardware platform. Leveraging the clustered model of the manycore platform, various scheduling models with different degrees of resource sharing can be implemented, ranging from partitioned approaches to more flexible cluster-based schedulers. Following this observation, ManyCOS, the PROXIMA manycore RTOS, has been designed as a generic RTOS framework, equipped with a plug-in mechanism that allows ManyCOS to assume different personalities. Scheduler plug-ins can be defined to realize a score of different configurations. In the scope of PROXIMA we plan to implement a scheduler plug-in for the following configurations:

- *Strictly partitioned scheduling*: where the RTOS (and its scheduling structures) is replicated on each core;
- *Global scheduling*: where the RTOS is responsible for scheduling the application globally over the physical cluster;
- *Global scheduling with master core*: similar to the previous configuration but with a core acting as a master and detaining the scheduler logic;

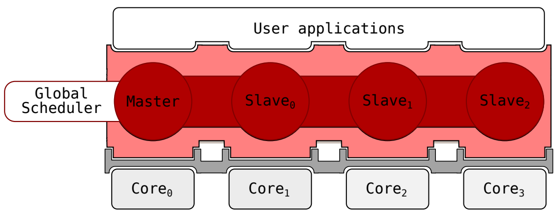
- *Cluster scheduling*: where scheduling is organized around exploiting configurable logical clusters within the physical one.



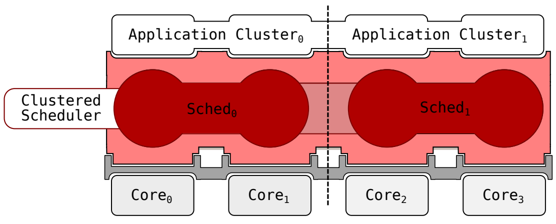
(a) Strictly partitioned.



(b) Global scheduling.



(c) Global scheduling with master core.



(d) Cluster scheduling.

Figure 2: Scheduling plug-in configurations.

The same configurations are depicted in Figure 2. The combination of highly configurable scheduling framework and HW-level mechanisms in the PROXIMA manycore platform allows users to choose among a wide range of configurations, depending on the applications to be deployed and their criticality level. As an example, a restrictive configuration would consist of deploying a high criticality application on a specific core within a cluster with strict partitioned scheduling and leave low criticality applications (soft real-time) to be scheduled globally on a separate cluster.
Time Composability, as defined in deliverable D6.3, is a property that applies to individual system components, both hardware and software: it guarantees that the bound on the timing behaviour of a component, in response to an execution request, can be determined independently of its composition with other components. A time-composable RTOS is notably expected to transparently support the execution of the user applications without incurring additional sources of history or data dependences [1,2]. The usefulness of a time-composable RTOS layer has been already evaluated in the scope of the PROARTIS project [9], in a single core setting with an ARINC-653 compliant RTOS [2]. Time composability in the RTOS layer was primarily realized by implementing constant-time kernel primitives, whose timing behaviour is independent from the SW state, and avoiding (or minimizing) any interference on the HW state that may affect the execution of the user applications. The same principles could be used in ManyCOS by leveraging on the various degree of segregation that can be guaranteed by the different instantiation of the RTOS framework, from completely partitioned to clustered systems.

3 Impact of platform design on the timing analyses

Current research on MBPTA has been focused on the multicore platform, and as such it is necessary to consider how it may be extended to the manycore platform. As discussed in D1.8, the primary difference between multicore and manycore is the presence of multiple inter-connected clusters. In MS2 D3.5, a simple argument has been constructed for the applicability of MBPTA to the BMP. Specifically, any delay caused by accesses to shared resources, be it interference or an arbitration delay, will have a probability distribution associated with when it occurs and for how long. Hence, any such latency is by definition, a random variable, and hence can be modelled more easily by MBPTA.

One important consideration is the manycore platform will suffer greater levels of interference from other tasks than the multicore. Therefore additional data and effort will be required to provide a similar level of confidence in the validity of the composability features than when compared with multicore. However, currently the effects of inter-task interference are not being considered. This will be addressed later in the project.

The taxonomy established in D3.1 distinguishes between intra-cluster and inter-cluster resources. The design changes introduced in D1.12 focus on an alternative inter-cluster communication infrastructure. The intra-cluster resources are left unchanged and the arguments presented in D3.5 hold true. We focus our discussion on inter-cluster resources and the mesh-based NoC.

Alternatives to the design of the PROXIMA manycore platform have been explored using a mesh-based infrastructure prevalent in HW COTS platforms. The proposed topology has been adapted to ensure the BMP continues to produce time-composable estimates, valid irrespective of the contention experienced by the analysed task, either through composable policies or the use of analysis-specific execution modes. This section discusses the impact of those alternatives in the design of the BMP and the applicability of MBPTA.

The BMP relies on a wormhole-based, meshed NoC to interconnect the different clusters. Arbitration of requests to each router's output port relies on a MBPTA-compliant randomised window policy. However to ensure the collected observations allow for the derivation of a composable timing estimate, valid irrespective of the contending tasks or their placement, the observed configuration should upper-bound the deployed one timing-wise.

The number of hops required for an analysed inter-cluster flow to reach its destination first impacts the number of experienced arbitration rounds, and thus the latency suffered by each packet. Using a XY routing algorithm, the number of hops traversed by a flow is maximised along the X and Y distances between its source and destination. Enforcing such a constraint during analysis, while observations of the analysed task are collected, should provide for composable estimates independent of the placement of the analysed flow.

The use of a wormhole-based protocol requires similar care to upper-bound the contention suffered by the analysed flow. Indeed an output port is dedicated to a packet until all its flits have been transmitted. Contention hence includes both direct and indirect effects. Direct contention corresponds to the case where the target output for the flow is allocated to an ongoing contending flow. Indirect contention occurs as a contending flow suffers from back-pressure, blocking the

analysed flow output port as it is itself block in a later hop.
The work-conserving arbitration policy exhibit a behaviour close to the arbitration policy in the intra-cluster tree with regards to contention. The computation of composable timing estimates relies on maximising during observations the number of pending requests by contenders in each traversed router. Such a behaviour can be achieved through hardware or software by ensuring all flows have the worst possible destination and emit request at the maximum allowed rate.

4 Inter-core Interference analysis

By its very design, the PROXIMA manycore platform can provide for the computation of time-composable estimates valid irrespective of the deployed system configuration. The performance of tasks are either isolated from each other, or the worst-case interference can be enforced during analysis. While isolation in the context of shared resource state is a strong requirement [6], enforcing worst-case interference is a difficult issue without hardware support and may result in pessimistic estimates, especially regarding inter-cluster communications.
The inter-core interference analysis, presented in depth in D3.8, builds a multivariate interference model that relates observed interferences to their impact on the execution time of the Unit of Analysis. The model effectively derives a safety margin specific to the contention observed on the inter-core shared resources. This margin is applied as a multiplicative factor on top of the execution time derived by an interference-free analysis, such as the PROXIMA timing analysis presented in Section 2.4 of deliverable D3.8. The inter-core interference analysis is a multi-step process that can be summarised as follows:

- Step 0. Build an instrumentation framework to capture an overview of the system's behaviour at runtime, through performance metrics beyond timings.
- Step 1. Select using Principal Components Analysis [4] the factors most relevant to the impact of interferences on the Unit of Analysis.
- Step 2. Build an ensemble [3] of interference models, gathered in an ensemble to forecast the impact of observed interference on the Unit of Analysis.
- Step 3. Derive an inflation factor for an observed inter-core interference scenario.

Step 0 and step 2, the construction of the instrumentation framework and inter-core interference model, are tied to the platform under analysis. The instrumentation relies on the platform monitoring infrastructure, e.g. performance monitoring counters. As discussed in Section 2, tracing on the simulator is extremely flexible and offers the collection of a large variety of statistics without incurring any additional cost on the Unit of Analysis. The interference model on the other hand should capture the setvs specific to the platform, deploying families of forecasting models appropriate to the setvs' properties.
The inter-core interference analysis can be applied at different levels on the BMP, either encompassing both intra- and inter-cluster resources in a single model, or deriving distinct models for each. The latter scenario allows the use of different forecasting models per family of resources. The safety margins computed for the intra- and inter-cluster interferences should be orthogonal and could be applied on top of each other.
The results of the inter-core interference analysis depend on the contention observed during Step 3, the final step of the analysis. The model can be fed a worst-case inducing interference scenario, but that would lead to the same pessimism as the use of the worst-case analysis mode. Changes in the contention suffered by the Unit of Analysis, unless they can be demonstrated to result in less interferences, require

the derivation of a new safety margin. Only the final step of the analysis should be reapplied; the interference model remains the same as long as the analysed task (or relevant parameters such as allocated partition in the cache) does not change. This is an additional argument for the separation of the intra- and inter-cluster interference models to enable the independent evaluation of the impact of changes on contention at different levels.
Development of the inter-core interference analysis, as discussed in D3.8, focused on the analysis of the Aurix/ERIKA COTS multi-core platform where mitigation strategies for inter-core interferences are limited. The BMP on the other hand allows by construction the computation of composable timing estimates. The deployment of the interference analysis on the BMP, subject to refinements of the instrumentation framework, still offers numerous benefits. The analysis indeed allows an evaluation and comparison of the composability, impact on interferences, and compliance to analysis of different architectural configurations. This also includes building an understanding of how timings within a cluster are impacted when inter-cluster interferences are considered. The evaluation of the manycore will provide guidelines for the design and analysis of manycore architectures. Such guidelines are an important requirement to establish with industrial partners a roadmap for the deployment of manycore platforms.

Evaluation of inter-core interferences on non-MBPTA compliant cluster

By its very design, the BMP allows the computation of composable timing estimates, valid irrespective of the interferences suffered by the analysed task at runtime. In this context, we use the interference model to provide an insight on the benefits of the PROXIMA many-core over a deterministic one. Because of the focus given to other platforms, the following experiments only present early results on a limited set of configurations and benchmarks. The synthetic contenders exercised during the interference analysis, as simple prototypes, offer only a limited range of interference levels each under a very regular access pattern.
We focus on the intra-cluster interferences resulting from the arbitration of concurrent access towards the shared memory. No software stack is used in the evaluation (see D2.10); each analysed application is ran directly on the target. The simulator is configured such that arbitration and caches in the 4-core cluster rely on deterministic policies, respectively round-robin arbitration and Least Recently User replacement policies. Two benchmarks, *matmult* and *dijkstra*, are considered in the experiments. *matmult* is a matrix multiplication algorithm with a very regular access pattern. *dijkstra* implements a shortest path search algorithm and its access pattern is driven by randomly generated input data.
The results of the application of the VICI analysis to each benchmark, under different input matrix or graph sizes, are presented in Table 1. For each configuration, we include in order the accuracy achieved by the analysis, the maximum prediction error in the test data, and the predicted interference multiplier. The regular access pattern in *matmult* is more amenable to a high accuracy result. As the manipulated matrices spill out of the cache at input size 32, more accesses need to traverse the intra-cluster tree to reach the main memory. The additional requests in addition to the cold misses inherent to the benchmark result in an increased sensitivity to interferences. Regarding *dijkstra*, the selected factors need to capture the impact of both inter-core interferences and input data such that the interference model can

distinguish between the two. When the input graph no longer fit into the cache, the additional accesses and delays further increase the gap between different input values.

Table 1: Accuracy and error of the VICI interference analysis applied to a deterministic many-core cluster.

Benchmark	Input size	Accuracy	Maximum Error	Interference multiplier
Matmult	26	97%	11.32%	1.1363
	32	88%	17.4%	1.5592
Dijkstra	8	87%	20.5%	1.4008
	16	90%	20.38%	1.4098
	32	32%	33.12%	1.0432

Those results corroborate with the factors selected by the analysis to model interferences. The simulator allows for the collection at no cost of a wide variety of statistics regarding the execution of the analysed task and its co-runners. 6 different factors are selected to model interferences for each application of the analysis reported in Table 1. The selected factors for both configurations of *matmult* relate to the accesses on the L2 cache from both the analysed core and the contenders. To distinguish between effects due to interferences or input data with a small graph, the analysis selects for *dijkstra* factors related to the private and shared cache usage. As the task sends more requests through the intra-cluster tree, the impact of interferences increases and interference-relevant factors prevail over local ones. But this selection of factors cannot allow for a distinction between internal and external variability.

5 Conclusions

In this deliverable, alternative designs to the platform and associated simulator have been considered in terms of their impact on the timing analysis to establish their validity. Both the PROXIMA manycore platform (D3.5) and its alternative design rely on the EVT analysis provided by INRIA for timing analysis. The philosophy behind the design of those platforms is to provide for the computation of time-composable estimates. The following is a review of how well this deliverable has met the requirements from deliverable D6.3. The text in *italics* is quoted text from D6.3.

1. From Manycore Simulator Success Criteria *techniques will be developed that allow timing behaviour to be modelled for that processor architecture, starting from the interconnect and then extending to the other features of interest.* - The relevant techniques have been developed and the BMP should by construction satisfy the requirements of the timing analyses. The quantitative evaluation of the relevant techniques is at best very preliminary and does not include a consolidated platform.

2. On Success Criteria - *The consolidated arguments and safety techniques/measures based on the results will be also ready.* - Arguments exist supporting the analysability of the BMP, but need to be supported by additional evaluation.

3. Support for WP1 - *Limitations in the capabilities of the hardware platforms of interest, in terms of tracing, debug and performance monitoring support, I/O mechanisms, PTA-compliance and interfacing* - Achieved

4. Support for WP4 - *Abstract model of the case studies characterising the applications, interactions, complexity and time behavior. Tool-chain requirements on instrumentation and timing analysis* - Achieved except for the abstract model of the use case. The capability has been created and is presented here, however the actual use case has not yet been analysed.

Acronyms and Abbreviations

- BMP: Bespoke Simulator Platform.
- COTS: Commercial Off-The-Shelf.
- EPC: Extended Path Coverage
- EVT: Extreme Value Theory.
- ETP: Execution time profile.
- HWM: High WaterMark
- MBPTA: Measurement-Based Probabilistic Timing Analysis.
- MCS: Mixed Criticality System
- NoC: Network on Chip.
- OS: Operating System
- PTA: Probabilistic Timing Analysis.
- pWCET: Probabilistic Worst-Case Execution Time.
- RVS: Rapita Verification Suite.
- setv: Sources of Execution Time Variability.
- VICI: Variable Inter-Core Interferences.
- WCET: Worst-Case Execution Time.

References

[1] A Baldovin, E Mezzetti, and T Vardanega. A Time-composable Operating System. In *Proceedings of the 12th International Workshop on Worst-Case Execution Time Analysis*, 2012.

[2] Andrea Baldovin, Enrico Mezzetti, and Tullio Vardanega. Towards a time-composable operating system. In *Proceedings of the 18th International Conference on Reliable Software Technologies - Ada-Europe 2013*, pages 143–160, 2013.

[3] S. V. Barai and Yoram Reich. Ensemble modelling or selecting the best model: Many could be better than one. *Artif. Intell. Eng. Des. Anal. Manuf.*, 13(5):377–386, November 1999.

[4] George H Dunteman. *Principal components analysis*. Number 69. Sage, 1989.

[5] J. Jalle, L. Kosmidis, J. Abella, E. Quinones, and F.J. Cazorla. Bus designs for time-probabilistic multicore processors. In *DATE*, 2014.

[6] Benjamin Lesage, Isabelle Puaut, and André Seznec. PRETI: Partitioned Real-time Shared Cache for Mixed-criticality Real-time Systems. In *Proceedings of the 20th International Conference on Real-Time and Network Systems*, RTNS '12, pages 171–180, New York, NY, USA, 2012. ACM.

[7] B. Nikolic, P. Meumeu Yomsi, and S.M. Petters. Worst-case memory traffic analysis for many-cores using a limited migrative model. In *19th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2013.

[8] M. Paolieri, E. Quinones, F.J. Cazorla, G. Bernat, and M. Valero. Hardware support for WCET analysis of hard real-time multicore systems. In *ISCA*, 2009.

[9] PROARTIS. Probabilistically analyzable real-time systems. feb 2010. http://www.proartis-project.eu/.

[10] M. Schoeberl, F. Brandner, J. Sparso, and E. Kasapaki. A statically scheduled time-division-multiplexed network-on-chip for real-time systems. In *NOCS*, 2012.

[11] Zheng Shi and A. Burns. Real-time communication analysis for on-chip networks with wormhole switching. In *Networks-on-Chip, 2008. NoCS 2008. Second ACM/IEEE International Symposium on*, pages 161–170, April 2008.