Grupo de Ingeniería de Software y Tiempo Real
Universidad de Cantabria

Modelling and Analysis Suite for Real Time Applications (MAST 1.6.0)

Analysis Techniques used in MAST

By: Michael González Harbour mgh@unican.es
José Carlos Palencia Gutiérrez palencij@unican.es
José Javier Gutiérrez García gutierjj@unican.es
Juan María Rivas Concepción rivasjm@unican.es

1. Introduction

The MAST toolset contains several schedulability analysis tools capable of analysing single processor and distributed systems scheduled with fixed priority, EDF, and EDF within priorities scheduling policies. The tools are based on different scheduling analysis techniques published in the literature:

- *Classic RM Analysis*. This analysis implements the classic exact response time analysis for single-processor fixed-priority systems first developed by Harter [7] and Joseph and Pandya [12], and later extended by Lehoczky to handle arbitrary deadlines [15] and by Tindell to handle jitter [34]. It corresponds to Technique 5, “Calculating response time with arbitrary deadlines and blocking”, in [14].
- *Varying Priorities Analysis*. This analysis implements the response time analysis for single processor fixed priority systems in which tasks may explicitly change their priorities, developed by González, Klein and Lehoczky [8]. It corresponds to Technique 6, “Calculating response time when priorities vary”, in [14]. In terms of the MAST model, transactions for this tool are built as linear transactions, with a sequence of activities, each representing the execution of an operation with a possibly different priority level specified through a *Permanent_Overriden_Priority* attribute. However, each transaction is limited to having a single segment. A segment is a continuous sequence of activities executed by the same server. Note that all the other fixed priority tools used in MAST require that the priority of an activity is the same at the start and at the end, and thus they do not allow using *Permanent_Overriden_Priority* attributes.
- *EDF Monopocessor Analysis*. This analysis implements the exact response time analysis for single-processor EDF systems first developed by Spuri [31]. In the MAST implementation we use the EDF Within Priorities (see below), because there may be interrupt service routines (modelled as fixed priority tasks) in addition to the EDF tasks.
- *EDF Within Priorities Analysis*. This analysis is a mixture of the response time analysis for fixed priority systems [14][15][34] and for EDF [31]. It is capable of analysing systems with hierarchical schedulers, in which the underlying primary scheduler is based on fixed priorities, and there may be other EDF (secondary) schedulers scheduling tasks at a given priority level. It was developed by González and Palencia [9].
- *Holistic Analysis*. This analysis extends the response time analysis to multiprocessor and distributed systems. It is not an exact analysis, because it makes the assumption that tasks of the same transaction are independent. It was first developed for fixed priority

systems by Tindell and Clark [34][35] and refined by Palencia et al [19]. For EDF systems, Spuri [32] and later Palencia [15] extended this technique for EDF systems scheduled with global-clock deadlines (referred to a global synchronized clock), and Rivas et al [37] for systems scheduled with local-clock deadlines (with local clocks in each processor).

- *Offset Based Approximate Analysis*. This is a response time analysis for multiprocessor and distributed systems that greatly improves the pessimism of the holistic analysis by taking into account that tasks of the same transaction are not independent, through the use of offsets. Offset based analysis for fixed priorities was first introduced by Tindell [36] and then extended to distributed systems by Palencia and González [20]. It was later extended to global-clock EDF systems by Palencia and González [22] and to local-clock EDF systems by Díaz de Cerio et al [5]. Although it provides much better results than the holistic analysis, it is not an exact method because the exact analysis is intractable. The method approximates the interference of a transaction with a maximum interference function that is independent on the phase of the transaction.

- *Offset Based Approximate with Precedence Relations Analysis*. This is an enhancement of the offset based approximate analysis for fixed priority systems in which the priorities of the tasks of a given transaction are used together with the precedence relations among those tasks to provide a tighter estimation of the response times. It was developed by Palencia and González [21], and later enhanced by Redell [25]. The extension to local-clock EDF was done by Díaz de Cerio et al [5].

- *Offset Based Slanted Analysis*. This is another enhancement of the offset based approximate analysis for fixed priority systems in which the maximum interference function is defined with a tighter approximation. This method provides better results that the Offset-Based Approximate Analysis, but it is uncertain if it gets better results than the method with precedence relations. In general both techniques should be applied, and the best results used as an upper bound on the response times, as both methods are pessimistic. It was developed by Mäki-Turja [18].

- *Offset Based Brute Force Analysis*. This is an exact analysis of offset-based transactions, initially developed by Tindell [36]. It analyses all possible combinations of tasks initiating the critical instant for each transaction, which leads to a combinatorial problem that only offers results for very small systems.

- *Analysis for heterogeneous distributed systems*. This technique [26] allows the integration of different response-time analysis techniques so that they can be applied to heterogeneous distributed systems with different scheduling policies in each resource. The Holistic and all the Offset-Based analysis techniques are now integrated in MAST into the distributed analysis for heterogeneous systems.

The MAST model is able to support multi-path transactions in which the event flow can be divided into several paths with fork (multicast) or branch (delivery or query server) elements, and paths may be joined with join (barrier) or merge (concentrator) elements. The analysis of these multipath transactions is described in [11] using the holistic analysis technique, and is implemented in MAST 1.5 with the following restrictions: the transaction has a single input event and only fork, join or merge elements are supported; branch is not supported, as the analysis in [11] only works for transactions with deadlines at or before the end of the period. The analysis of multi-path transactions with offset-based techniques was published by Amurrio et al [1][2], and implemented in MAST 1.6.

The MAST toolset is able to automatically calculate the blocking times caused by mutual exclusion synchronization. The model includes shared resources with the immediate ceiling [4], priority inheritance [29], and stack resource [4] synchronization protocols. It also allows mixtures of these protocols and their use in multiprocessor systems [23][24], with some restrictions that are described below. For the basic priority inheritance, Rodríguez and García [27] showed that most implementations do not strictly follow the rules in [29], and that the amount of blocking is usually higher than that predicted by the theory. In MAST we take this into account when calculating the blocking times due to the use of the priority inheritance protocol.

The MAST toolset also contains tools to automatically assign priorities and other scheduling parameters. Priority assignment tools are provided for single-processor and distributed systems. In single-processor systems, if deadlines are within the periods the optimum deadline monotonic priority assignment developed by Leung and Layland is used [16]. The Liu and Leyland classic rate monotonic priority assignment for the case of deadlines equal to the periods [17] is know to be a special case of the deadline monotonic assignment. When deadlines are larger than the task periods, the optimum priority assignment developed by Audsley is used [3]. This technique is based on the iterative use of the schedulability analysis tools for different solutions, until a schedulable solution is obtained.

In multiprocessor and distributed systems the problem of assigning scheduling parameters is much harder, as there are strong interrelations between the response times in the different resources. For distributed systems scheduled with fixed priorities, we provide two heuristic solutions based on iteratively applying the schedulability analysis tools. The first one is based on the use of the simulated annealing optimization techniques, first used by Tindell, Burns, and Wellings for assigning priorities [33]. The second heuristic, which usually provides better and faster results is the HOPA algorithm developed by Gutiérrez and González [10]. For distributed EDF systems, the HOSDA [37] algorithm, which is en evolution of HOPA, is provided. This algorithm is capable of assigning and optimizing local and global scheduling deadlines. In addition to HOSDA, other simpler mechanisms algorithms are provided for comparison purposes [42]: the Proportional Deadline assignment (PD), the Normalized Proportional Deadline assignment (NPD) , the Ultimate Deadline (UD) and the Effective Deadline (ED). These algorithms were interpreted to make them suitable for allocating both priorities and scheduling deadlines in [38][39]. In addition, we have implemented the Equal Slack (EQS) and Equal Flexibility (EQF) algorithms originally proposed in [13] and interpreted to make the suitable for an off-line priority or scheduling deadline assignment in [40]. All these techniques can be applied for the assignment of both local [41] and global [37] scheduling deadlines. All these techniques are integrated into HOSPA [26], the scheduling parameters assignment tool provided for heterogeneous FP/EDF systems.

The MAST toolset is able to determine not only whether the system is schedulable or not, but also how far it is from being schedulable, or how much capacity is available until the system becomes unschedulable. It does so by calculating slacks, which are defined as the percentage by which we can increase the execution times of some operations while keeping the system schedulable (for positive slacks) or the percentage by which we have to decrease the execution times to make the system schedulable (for negative slacks). A slack of zero means that the system is just schedulable, and that even the smallest increment in the execution times would lead to non schedulability. There are different kinds of slacks provided:

- *System slack*: affects all the operations in the system

- *Processing resource slack*: affects only the operations executed in a given processing resource
- *Transaction slack*: affects only the operations used in a given transaction
- *Operation slack*; affects only one single operation

The slacks are calculated by modifying the worst-case execution times and repeating the analysis using a binary search to find the point in which the system becomes unschedulable (or schedulable if it wasn't). The slack is calculated with a 1% precision to limit the amount of times the analysis is repeated to around 20 times.

2. Architecture of Mast analysis tools

2.1 mast_analysis.adb:

This is the main program in the analysis tool. These are the main steps that it performs:

- Processes the command used to launch Mast with its many options and selects the analysis tool to be used
- Parses the input file, previously converting it to text format if it is an XML file
- Performs the consistency checks
- Performs the scheduling parameters assignment if requested
- Calculates slacks if requested
- Performs the final schedulability analysis
- Writes the results to the output file

The selection of the analysis tool is stored in these variables, for later processing:

- `The_Tool`: A pointer to the analysis tool found in mast-tools.ads
  - In case of a distributed analysis (holistic or one of the offset-based analysis), `The_Tool` always points at `mast.tools.Distributed_Mixed_Analysis` and stores the requested analysis in the variable `User_Analysis`, for later processing.
  - In case the selected tool is "default", `The_Tool` always points at `mast.tools.Default_RTA_Analysis` and `User_Analysis` is set to `offset_based_slanted`
  - In case the selected tool is "offset_based", `The_Tool` always points at `mast.tools.Distributed_Mixed_Analysis` and `User_Analysis` is set to `offset_based_slanted`
- `Requested_Analysis_Tool`: stores the actual tool requested by the user, for later reporting

2.2 mast-tools.ads

This package contains the drivers to the mast analysis tools. There is a special driver that invokes one of the other drivers according to the system's configuration:

- `Default_RTA_Analysis`. Uses the default technique for the system:

  if system is Monoprocessor_Only and Simple_Transactions_Only then

  - `Classic_RM_Analysis` for fixed priorities
  - `EDF_Monoprocessor_Analysis` for EDF
  - `EDF_Within_Priorities` for hierarchical EDF on top of fixed priorities

  else

  - `Distributed_Mixed_Analysis` with `User_Analysis` = `Offset_Based_Approx` if system is EDF
  - `Distributed_Mixed_Analysis` with `User_Analysis` = `Offet_Based_Slanted` if system is FP

The other drivers check the tool-related restrictions on the system, calculate blocking times, and then invoke the actual analysis tool. These drivers are:

- Single processor tools
  - `Classic_RM_Analysis`: for Fixed Priority systems
  - `Varying_Priorities_Analysis`: For Fixed Priority systems in which tasks explicitly change their priorities
  - `EDF_Monoprocessor_Analysis`: For EDF systems. In practice the analysis is done with the `EDF_Within_Priorities` technique
  - `EDF_Within_Priorities_Analysis`: For hierarchically scheduled systems with a fixed priority scheduler at the bottom and an EDF secondary scheduler executing at the specified priority level
- Distributed tools
  - `Distributed_Mixed_Analysis`:
    - If the system has linear transactions it invokes `mast-linear_task_analysis_tools.Distributed_RTA`
    - If the system is multipath it invokes the `multipath_analysis` driver
  - `Multipath_Analysis`: it invokes `mast-linear_task_analysis_tools.Distributed_RTA`

In addition, the package contains the following tools:

- Scheduling parameters assignment tools:
  - `Monoprocessor_Assignment`
  - `Linear_HOSPA`
  - `Linear_Simulated_Annealing_Assignment`
  - `Multipath_HOPA`
  - `Multipath_Simulated_Annealing_Assignment`
- Shared Resource Analysis Tools

  - `Calculate_Ceilings_And_Levels`
  - `Calculate_Blocking_Times`
- Miscellaneous Tools
  - `Utilization_Test`
  - `Check_Shared_Resources_Total_Ordering`
  - `Check_Transaction_Schedulability`
  - `Check_System_Schedulability`
  - `Calculate_Processing_Resource_Utilization`
  - `Calculate_System_Utilization`

2.3 mast-linear_task_analysis_tools

This package contains the response time analysis for heterogeneous systems in which each processing resource may be scheduled by a different scheduling policy. It contains the following operations:

- `Initialize_Processor_Analysis_Accesses`: an operation to initialize the analysis tools to be used for each processing resource.
- `Distributed_RTA`: a general driver to the task analysis. It iterates the analysis until a stopping condition is met. For each analysis iteration it iterates over all the transactions and all the tasks. For each task it invokes its corresponding analysis tool (see below) and then propagates the results to the next task(s).
- Analysis tools, that make a partial analysis for a particular task in a particular transaction:
  - `Holistic_Task_FP`
  - `Holistic_Task_EDF_Local`
  - `Holistic_Task_EDF_Global`
  - `Offset_Based_approx_w_pr_Task_FP`
  - `Offset_Based_approx_Task_FP`
  - `Offset_Based_Slanted_Task_FP`
  - `Offset_Based_Brute_Force_Task_FP`
  - `Offset_Based_approx_Task_EDF_Local`
  - `Offset_Based_approx_w_pr_Task_EDF_Local`
  - `Offset_Based_approx_Task_EDF_Global`

2.4 Mast.Monoprocessor_Tools

This package contains the single processor analysis tools, invoked from the drivers in `Mast.Tools`. These tools are:

- Fixed priority analysis and priority assignment tools
  - RM_Analysis

    - Varying_Priorities_Analysis
    - Priority_Assignment
- EDF analysis and scheduling deadline assignment tools
    - EDF_Monoprocessor_Analysis
    - EDF_Within_Priorities_Analysis
    - Deadline_Assignment

2.5 Mast.Miscellaneous_Tools

- Shared resources tools
    - Calculate_Ceilings_And_Levels
    - Calculate_Blocking_Times
- Schedulability tests
    - Utilization_Test
    - Check_Transaction_Schedulability
    - Check_System_Schedulability
    - Calculate_Processing_Resource_Utilization
    - Calculate_System_Utilization
    - Calculate_Average_Response_Times

References

[1] Andoni Amurrio, Ekain Azketa, J. Javier Gutierrez, Mario Aldea, and Michael González Harbour, “Response-Time Analysis of Multipath Flows in Hierarchically-Scheduled Time-Partitioned Distributed Real-Time Systems”. IEEE Access, vol. 8, pp. 196700-196711, November 2020. https://doi.org/10.1109/ACCESS.2020.3033461

[2] Andoni Amurrio González, and J. Javier Gutiérrez and Mario Aldea (Thesis Supervisors). “Schedulability Analysis and Optimization of Time-Partitioned Distributed Real-Time Systems”. PhD. Thesis, University of Cantabria, March 2022. http://hdl.handle.net/10902/24733

[3] N.C. Audsley, “Optimal Priority Assignment and Feasibility of Static Priority Tasks with Arbitrary Start Times”, Dept. Computer Science, University of York, December 1991.

[4] T.P. Baker, “Stack-Based Scheduling of Realtime Processes”, Journal of Real-Time Systems, Volume 3, Issue 1 (March 1991), pp. 67–99.

[5] U. Díaz De Cerio, M. González Harbour, J.C. Palencia, and J.P. Uribe. “Adding Precedence Relations to the Response-Time Analysis of EDF Distributed Real-Time Systems”. 22nd International Conference on Real-Time Networks and Systems, RTNS 2014, October, 2014.

[6] U. Díaz-de-Cerio, J.P. Uribe, M. González Harbour and J.C. Palencia. “On the convergence of the holistic analysis for EDF distributed systems”. Journal of Systems Architecture, Volume 61(9), pp.398–409, Elsevier, October 2015.

[7] P.K. Harter, “Response times in level-structured systems”. ACM Transactions on Computer Systems, vol. 5, no. 3, pp. 232-248, Aug. 1984.

[8] M. González Harbour, M.H. Klein, and J.P. Lehoczky. “Timing Analysis for Fixed-Priority Scheduling of Hard Real-Time Systems”. IEEE Trans. on Software Engineering, Vol. 20, No.1, January 1994.

[9] M. González Harbour and J.C. Palencia, “Response Time Analysis for Tasks Scheduled under EDF within Fixed Priorities”, Proceedings of the 24th IEEE Real-Time Systems Symposium, Cancun, México, December, 2003.

[10] J.J. Gutiérrez García and M. González Harbour, “Optimized Priority Assignment for Tasks and Messages in Distributed Real-Time Systems”. Proceedings of the 3rd Workshop on Parallel and Distributed Real-Time Systems, Santa Barbara, California, pp. 124-132, April 1995.

[11] J. J. Gutiérrez García, J.C. Palencia Gutiérrez and M. González Harbour, “Schedulability Analysis of Distributed Hard Real-Time Systems with Multiple- Event Synchronization”. Proceedings of 12th Euromicro Conference on Real-Time Systems, Stockholm (Sweden), IEEE Computer Society Press, pp. 15-24, June 2000.

[12] M. Joseph and P. Pandya, “Finding Response Times in a Real-Time System”. The Computer Journal (British Computing Society) 29, 5, pp. 390-395, October 1986.

[13] Kao H., Garcia-Molina H., “Deadline assignment in a distributed soft real-time system”. Proceedings of the 13th International Conference on Distributed Computing Systems, IEEE (1993), pp. 428-437 .

[14] M.H. Klein, T. Ralya, B. Pollak, R. Obenza, and M. González Harbour. “A practitioner’s Handbook for Real-Time Analysis”. Kluwer Academic Pub., 1993.

[15] J.P. Lehoczky. “Fixed Priority Scheduling of Periodic Tasks Sets with Arbitrary Deadlines”. IEEE Real-Time Systems Symposium, 1990.

[16] J. Leung, and J.W. Layland. “On Complexity of Fixed-Priority Scheduling of Periodic Real-Time Tasks”. Performance Evaluation 2, 237-50, 1982.

[17] C.L. Liu and J.W. Layland. “Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment”. Journal of the ACM, Vol. 20, No. 1, 1973, pp. 46-61.

[18] Jukka Mäki-Turja and Mikael Nolin. “Efficient implementation of tight response-times for tasks with offsets”. Journal of Real-Time Systems. Volume 40 Issue 1, October 2008, pp. 77 - 116.

[19] J.C. Palencia Gutiérrez, J.J. Gutiérrez García, M. González Harbour. “On the schedulability analysis for distributed hard real-time systems”. 9th Euromicro Workshop on Real-Time Systems. Toledo, 1997.

[20] J.C. Palencia Gutiérrez and M. González Harbour, "Schedulability Analysis for Tasks with Static and Dynamic Offsets". Proceedings of the 18th. IEEE Real-Time Systems Symposium, Madrid, Spain, December 1998.

[21] J.C. Palencia, and M. González Harbour, “Exploiting Precedence Relations in the Schedulability Analysis of Distributed Real-Time Systems”. Proceedings of the 20th IEEE Real-Time Systems Symposium, 1999.

[22] J.C. Palencia and M. González Harbour, “Offset-Based Response Time Analysis of Distributed Systems Scheduled under EDF”. Euromicro conference on real-time systems, Porto, Portugal, June 2003.

[23] R. Rajkumar, L. Sha, and J.P. Lehoczky. “Real-Time Synchronization Protocols for Multiprocessors”. IEEE Real-Time Systems Symposium, December 1988.

[24] R. Rajkumar. “Real-Time Synchronization Protocols for Shared Memory Multiprocessors”. Proceedings of the 10th International Conference on Distributed Computing, 1990.

[25] Redell, O. Response Time Analysis for Implementation of Distributed Control Systems, Doctoral Thesis, TRITA-MMK 2003:17, ISSN 1400-1179, ISRN KTH/MMK/R--03/17--SE, 2003

[26] Juan M. Rivas, J. Javier Gutiérrez, J. Carlos Palencia, and Michael González Harbour. “Schedulability Analysis and Optimization of Heterogeneous EDF and FP Distributed Real-Time Systems”. ECRTS-2011, Porto, June 2011.

[27] P. Rodríguez Hernández and J.J. García Reinoso, “Nota sobre la Implementación del mecanismo de herencia”, VI Jornadas de Tiempo Real, Gijón, February 2003 (In Spanish).

[28] L. Sha, T. Ralya, and J.B. Goodenough, “An analytical approach to real-time software engineering”, Software Engineering Institute Annual Technical Review, 1989.

[29] L. Sha, R. Rajkumar, and J.P. Lehoczky. “Priority Inheritance Protocols: An approach to Real-Time Synchronization”. IEEE Trans. on Computers, September 1990.

[30] B. Sprunt, L. Sha, and J.P. Lehoczky. “Aperiodic Task Scheduling for Hard Real-Time Systems”. The Journal of Real-Time Systems, Vol. 1, 1989, pp. 27-60.

[31] M. Spuri. “Analysis of Deadline Scheduled Real-Time Systems”. RR-2772, INRIA, France, 1996.

[32] M. Spuri. “Holistic Analysis of Deadline Scheduled Real-Time Distributed Systems”. RR-2873, INRIA, France, 1996.

[33] K.W. Tindell, A. Burns, and A.J. Wellings. “Allocating Real-Time Tasks. An NP-Hard Problem Made Easy”. Real-Time Systems Journal, Vol. 4, No. 2, May 1992. pp. 145- 166.

[34] K. Tindell, “An Extendible Approach for Analysing Fixed Priority Hard Real-Time Tasks”. Journal of Real-Time Systems, Vol. 6, No. 2, March 1994.

[35] K. Tindell, and J. Clark, “Holistic Schedulability Analysis for Distributed Hard Real-Time Systems”. Microprocessing & Microprogramming, Vol. 50, Nos.2-3, pp. 117-134, April 1994.

[36] K. Tindell, “Adding Time-Offsets to Schedulability Analysis”, Technical Report YCS 221, Dept. of Computer Science, University of York, England, January 1994.

[37] J.M. Rivas, J.J. Gutiérrez, J.C. Palencia, and M. González Harbour, “Optimized Deadline Assignment and Schedulability Analysis for Distributed Real-Time Systems with Local EDF Scheduling”. Proceedings of the 2010 International Conference on Embedded Systems and Applications, Las Vegas, Nevada, USA, ESA'2010. CSREA Press, July, 2010, ISBN:1-60132-141-4, pp. 150,156.

[38] Juan María Rivas Concepción and J. Javier Gutiérrez (Thesis Supervisor). “Estudio de la planificabilidad y optimización de sistemas distribuidos de tiempo real basados en prioridades fijas y EDF. PhD. Thesis, University of Cantabria, July 2015.

[39] Juan M. Rivas, J. Javier Gutiérrez, J. Carlos Palencia, and Michael González Harbour. “Deadline Assignment in EDF Schedulers for Real-Time Distributed Systems”. IEEE Transactions on Parallel and Distributed Systems, Vol. 26(10), pp. 2671-2684, October 2015.

[40] Juan M. Rivas and J. Javier Gutiérrez. “Interpretación de dos algoritmos EDF on-line para la optimización de sistemas distribuidos de tiempo real”. V Simposio de Sistemas de Tiempo Real in the V Congreso Español de Informática, CEDI 2016, Salamanca (Spain), September 2016.

[41] J. Sun, “Fixed Priority scheduling of end-to-end periodic tasks”, Ph.D. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1997.

[42] J. Liu, “Real-Time Systems,” Prentice Hall, 2000.