1. Quick Overview

GEN4MAST is a tool to massively generate synthetic real-time systems descriptions, and test real-time techniques on them (schedulability analysis, priority assignment, etc.). It's built around the MAST suite of tools:

- The systems are described using the MAST System Model [1][2]. This model uses the principles described in the MARTE standard [3].
- The real time techniques that are applied to the generated systems are the ones included in the MAST Analysis Tool. This tool is open source, and new techniques can be added, granted they are compatible with the MAST System Model.

GEN4MAST also integrates the results in a database, and provides functions to access and present these results with graphics, tables, etc. The execution of GEN4MAST is configured with an input file, which has a custom syntax. This file defines the characteristics of the synthetic pool of examples to be generated, and which real-time techniques to apply on them. GEN4MAST supports the execution in a supercomputer that uses the TORQUE resource management system. This support depends on the characteristics of the supercomputer used. The source code of GEN4MAST implementation can be found in "include/cluster.py".

1.1. System Generation

The GEN4MAST configuration file contains the generation parameters that describe the characteristics of the systems to be generated. The user can configure for example the number of end-to-end flows, steps in each end-to-end flow, periods, number of processors, etc. GEN4MAST can generate system with different characteristics at the same time. The pool of systems are generated taking into consideration Lehoczky's Breakdown Utilization criterion [4], or maximum schedulable utilization. For each combination of generation parameters, a Base System is defined. The worst case execution times in this base system are sequentially increased to form Utilization Series. Therefore, the pool of generated examples is composed of several Utilization Series. A description of the generation parameters is given in the Section 3 of this document.

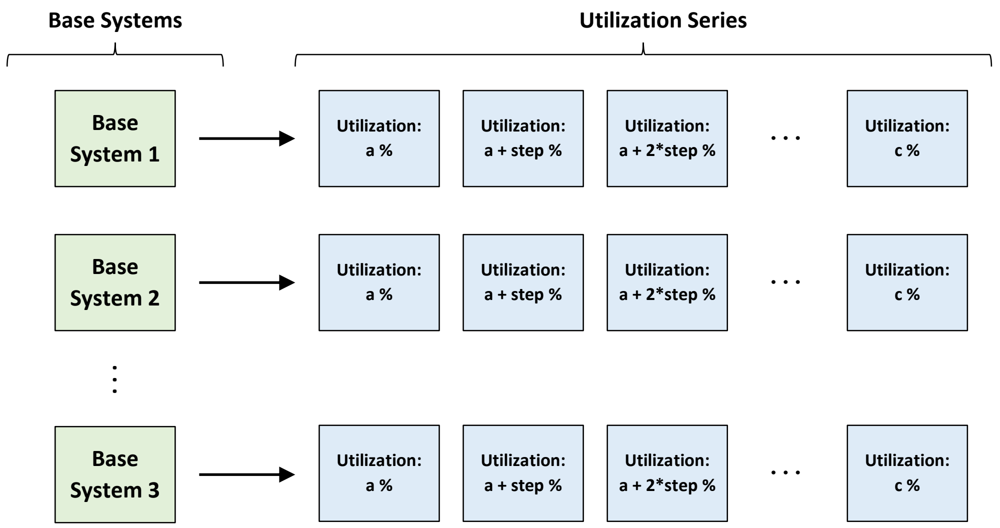

Figure 1 Pool of examples created by GEN4MAST, with Utilization Series composed of systems with utilizations ranging from a% to c%.

1.2.Real time techniques execution

The GEN4MAST configuration file also contains the specification of which real time techniques should be applied to the pool of generated systems. These are the techniques implemented in the MAST Analysis Tool (mast_analysis.exe) (www.mast.unican.es). The kind of techniques whose application can be automated with GEN4MAST are:

- Schedulability Analysis Techniques, which calculate the worst case response times to determine the schedulability of the systems.
- Optimization Techniques, which calculate an assignment of scheduling parameters, that is, fixed priorities for FP or scheduling deadlines for EDF, to maximize the schedulability of the system.
- Sensitivity Analysis through System Slack, which is the percentage by which all the steps execution times can be increased while keeping the system schedulable.

A description of the techniques supported in GEN4MAST is given in the Section 3 of this document.

2. Quick Tutorial

This quick tutorial describes step by step the process of performing a simple study using GEN4MAST, from the definition of the GEN4MAST configuration file, to the creation of graphics from the results.

This tutorial uses the GEN4MAST binary for windows (inside the “build_exe” directory). This binaries were created from the GEN4MAST python scripts using *pyinstaller*. For other operating systems, the GEN4MAST python scripts can be used directly.

To perform this tutorial, you’ll need:

- GEN4MAST. In this tutorial the binaries for windows are used.
- Python 2.7.
- Pytables.
- Matplotlib.

It’s a good idea to add GEN4MAST executables into the PATH of your system. In this tutorial we’ll assume this is the case.

In the “tutorial” directory you can find the GEN4MAST configuration file “config.txt” for this tutorial. In the study represented in this configuration file, we want to observe how the worst case response times, and maximum schedulable utilization, using two different response time analysis techniques, are influenced by the number of steps in the end-to-end flows. This configuration file contains the following lines, whose configuration parameters are further explained in Section 3 of this document:

`[GENERATOR]`	`Section in which the characteristics of the systems to be generated are specified.`
`POPULATION 4`	`The process is repeated 4 times, to obtain more statistically relevant results.`
`N_FLOWS 4`	`Systems with 4 end-to-end flows.`
`N_STEPS 5 7 9`	`Systems whose end-to-end flows will have 5, 7 and 9 steps.`
`SINGLE_FLOWS 0`	`0% of end-to-end flows with 1 step (independent tasks).`
`FIXED_LENGTH True`	`Every end-to-end flow have the same number of steps.`

N_PROCESSORS 3	The systems have 3 processors.
REPETITION YES	The end-to-end flows can cross the same processor more than once.
SCHEDULING_POLICY FP	Processors scheduled by FP policy (Fixed Priorities)
PERIOD_BASE 10	
PERIOD_DISTRIBUTION LOG-UNIFORM	
PERIOD_RATIO 100	End-to-end flows periods selected in the range [10, 100*100], using a logarithmic uniform distribution.
UTILIZATION_START 10	
UTILIZATION_STEP 2	
UTILIZATION_TOP 98	Series of utilizations in the range 10% to 98%, in 2% steps.
UNIFORM_UTILIZATION True	Every processor have always the same utilization.
WORKLOAD UUNIFAST SCALE-WCET	Worst case execution times of the steps calculated with two methods, UUnifast, and SCALE-WCET.
BEST_CASE 0	Best case execution times of the steps equal to 0% of the worst case execution times.
DEADLINES NT	End-to-end deadlines equal to N*T, where N is the number of steps of the end-to-end flow, and T its period.
[/GENERATOR]	End of the GENERATOR section.
[EXECUTION]	Start of the EXECUTION sector, where the tools to be applied to the generated systems are specified.
MAST_PATH C:\G4MAST\mast_analysis_b.exe	Path to the special MAST tool (included with GEN4MAST, inside the "mast_analysis" directory).
ANALYSIS_TOOL HOLISTIC OFFSET SLANTED	Response time analysis tools to be applied: Tindell's Holistic analysis, and Tindell's Offset Based approximate analysis.
DEADLINES_TYPE GLOBAL	Scheduling deadlines to be considered global (only for EDF, not applied in this example)
ASSIGNMENT_TOOL PD	Fixed priorities assigned using the PD algorithm
[/EXECUTION]	End of the EXECUTION section.

To perform this study, we can execute this command in the "tutorial" directory, where "config.txt" resides, assuming "generate.exe" is in the system PATH (for windows):

```
generate config.txt –g –e –l
```

This command generated the systems ("-g"), the MAST execution scripts ("-e"), and launches the executions ("-l"). If not using Windows, the "generate.py" script should be used instead (assuming that its directory is included in the system PATH):

```
generate.py config.txt –g –e –l
```

Once the "generate" command has finished, four new folders (1, 2, 3 and 4) will appear, one for each repetition of the process to obtain statistically relevant results (the POPULATION 4 parameter in the configuration file). These four directories contain the systems, scripts and results of the study. The next step is to compile the results into an indexed database, for easier access to the results. For this, this command can be performed:

```
storeh5 . data.h5
```

This command search recursively for results in the current directory ("."), and stores them in a database ("data.h5"). This database uses the Pytables format, and can be accessed using the Vitables tool (http://vitables.org/). GEN4MAST provides python methods to easily create figures from these results. In this tutorial, a python script is provided that created three figures from the results ("graphics.py"). Before executing this script, its second line ("sys.path.append()") must be modified to point to the location of your own GEN4MAST root directory

This script can be executed with this command:

```
python graphics.py
```

This scripts generates 3 figures. "results_bars.png" is a bar chart with the maximum schedulable utilization reached by each Response Time Analysis technique and systems with each end-to-end flow length. "result_lines_HOLISTIC.png" and "result_lines_OFFSET.png" represent the average worst case response times for each end-to-end flow length, for the holistic and offset based analysis respectively.

This tutorial provides a quick overview of the capabilities and usage of GEN4MAST. In the next section, a complete description of the GEN4MAST configuration file, with all its available options, is provided

3. Configuration File Specification

The GEN4MAST configuration file is a plain text file. Three different optional sections are defined: GENERATOR, ARCHITECTURE and EXECUTION. The sections include a series of parameters for which different values can be set. The study will be the product of the combination of all these values.

In the GENERATOR and ARCHITECTURE sections, the characteristics of the systems to be generated are defined. The GENERATOR section sets the system generation rules, for example, the rule by which the periods are generated. On the other hand, the ARCHITECTURE section is used to specify the actual characteristics of the systems, for example the actual values of the periods.

The EXECUTION section includes the definition of which real time techniques to apply to the generated systems (RTA technique, priority assignment, etc.). A description of the parameters follows.

GENERATOR SECTION

[GENERATOR]	Section start.
POPULATION	Integer. Number of times the generation process is repeated to generate statistically relevant results.
N_FLOWS	Integer. Number of end-to-end flows.
N_STEPS	Integer.
SINGLE_FLOWS	Integer.
FIXED_LENGTH	Boolean. If FIXED_LENGTH is **True**, every end-to-end flow have N_STEPS steps. If **False**, the length of every end-to-end flow is uniformly selected in the range [2,N_STEPS]. MONO_FLOWS states the percentage of end-to-end flows that have only one step (independent tasks).

N_PROCESSORS	Integer. Number of processors available in the system. Steps are statically assigned a processor. This initial version of GEN4MAST only generates processors, not networks. All the same, networks are usually analyzed like another processor in the system, with certain caveats.
REPETITION	String. Accepted values: **NO**: End-to-end flows should not cross the same processor more than once (only possible if N_PROCESSORS is equal or larger than the length of the end-to-end flow). **YES**: The processor of each step is randomly selected in the range [1,N_PROCESSORS].
SCHEDULING_POLICY	String. Accepted values: **FP**: Processors are scheduled by a Fixed Priorities policy. **EDF**: Processors are scheduled by an Earliest Deadline First policy.
PERIOD_BASE	Integer.
PERIOD_RATIO	Integer.
PERIOD_DISTRIBUTION	String. End-to-end flows periods are randomly selected in the range [PERIOD_BASE, PERIOD_BASE*PERIOD_RATIO]. Two different probability distributions are provided are set with PERIOD_DISTRIBUTION, with accepted values: **CUSTOM-UNIFORM:** Uniform distribution, but first and second end-to-end flows have the maximum and minimum period in the range respectively; **LOG-UNIFORM**: A logarithmic uniform distribution.
DEADLINES	String or Integer. End-to-end deadlines. Can be an integer K, so that D=K*T, where T is the period of the end-to-end flow. Also, it can specify a point in the segment [T,N*T], where N is the length of the end-to-end flow, with accepted values: **T** and **NT**: the extremes of the segment; **T1** and **T2**: the first and second thirds of the segment; **Q1**, **Q2** and **Q3**: the first, second and third quarter of the segment.
UTILIZATION_START	Integer.
UTILIZATION_STEP	Integer.
UTILIZATION_TOP	Integer. The utilization series are generated in the range [UTILIZATION_START,UTILIZATION_TOP], with steps of UTILIZATION_STEP.
UNIFORM_UTILIZATION	Boolean. If True, every processor has always the same utilization. If False, the processors can have different utilizations, averaging the same system utilization.
WORKLOAD	String. Method to generate the worst case execution times of the steps. Accepted values: **UUNIFAST** applies the UUnifast method [5], with uniform step utilization; **SCALE-WCET**: every step have the same step utilization.
BEST_CASE	Integer. The steps have best case execution times as BEST_CASE percentage of the worst case execution times.
[GENERATOR]	Section end.

ARCHITECTURE

[ARCHITECTURE]	Section start.
LENGTH	A Python list that specifies the number of steps in each end-to-end flow For example, [2,4,5] specifies a system with 3 end-to-end flows, the first one with 2 steps, the second one with 4 steps, and the third one with 5 steps.
PROCESSORS	A Python list with N inner lists, one for each end-to-end flow in the system. It specifies the processor in which each step is located. For example, [[1,2],[4,3,1,2],[1,4,2,3,1]] sets the localization of the steps in 3 end-to-end flows, in which the 2 steps of the first end-to-end flow are located in the processor 1 and 2 respectively.

PERIODS	A Python list that specifies the periods of the end-to-end flows. For example, [100,150,340] specifies the periods of 3 end-to-end flows, in which the first one has a period of 100, the second of 150, and the third of 340.
DEADLINES	A Python list that specifies the end-to-end deadlines of the end-to-end flows. For example, [300,400,450] sets the end-to-end deadlines of 3 end-to-end flows, in which the first one has an end-to-end deadline of 300, the second one of 400, and the third one of 450.
[/ARCHITECTURE]	Section end.

EXECUTION

[EXECUTION]	Section start.
MAST_PATH	String. Absolute path of the MAST analysis tool. The initial version of GEN4MAST needs a modified version of this executable ("mast_analysis_b.exe"), provided in the "mast_analysis" directory of GEN4MAST.
ANALYSIS_TOOL	String. Response Time Analysis technique to apply. Accepted values: **HOLISTIC**: Holistic analysis [6]; **OFFSET**: Offset-Based analysis [7]; **SLANTED**: Offset-Based Slanted analysis [8]; **OFFSET_OPT**: Offset-Based analysis with precedence relationships optimizations [9]; **BRUTE_FORCE**: Offset-Based Brute Force analysis [7].
DEADLINES_TYPE	String. For EDF scheduling, it specifies if the scheduling deadlines should be considered as local or global. Accepted values: **LOCAL**: Local scheduling deadlines, for LC-EDF scheduling [10]; **GLOBAL**: Global scheduling deadlines, for GC-EDF scheduling with clock synchronization [10].
ASSIGNMENT_TOOL	String. Specifies the scheduling parameters assignment tool to apply (fixed priorities or scheduling deadlines). Accepted values: **UD [11]**, **ED [11]**, **PD [12]**, **NPD [12]**, **EQS [11]**, **EQF [11]**, **HOSPA [13]**.
STOP_FACTOR	Integer. Specifies the analysis stop factor. If during the analysis, any worst case response time surpasses D*STOP_FACTOR, the analysis is stopped, where D is the end-to-end deadline. If the analysis is stopped, the system is deemed non schedulable, and the worst case response times set to infinite.
SLACK	Boolean. If se to True, the System Slack is calculated.
JITTER_AVOIDANCE	Boolean. If True, the analysis is performed emulating the elimination of all the release jitters caused by the precedence relationships of the steps.
LC-EDF-GSD	Boolean. If True, the LC-EDF-GSD modification is applied to the local scheduling deadlines assignment [10].
LC-EDF-DS-FACTOR	Integer. States the K factor in the LC-EDF-DS modification in the assignment of local scheduling deadlines [10].
[/EXECUTION]	Section start.

The execution of HOSPA can be configured with the following parameters [13]:

K	Pair if floats separated by comma. Sets the k pair for the HOSPA algorithm.
ITERATIONS_PER_PAIR	Integer. Sets the number of iterations to be performed for each k pair.
OPTIMIZATION_ITERATIONS	Integer. Sets the number of iterations to be performed over an already schedulable solution, to further optimize it.

4. Contact Information

Juan Maria Rivas Concepcion

Software engineering and real-time group

Universidad de Cantabria

Avda. de Los Castros s/n

39005 - Santander

Spain

e-mail:

rivasjm@unican.es

jrivcon@gmail.com

5. References

[1] M. González Harbour, J.J. Gutiérrez, J.C. Palencia and J.M. Drake, “MAST: Modeling and Analysis Suite for Real Time Applications,” Proc. of the 13th Euromicro Conference on Real-Time Systems, Delft (The Netherlands), pp. 125-134, 2001.

[2] M. González Harbour, J.J. Gutiérrez, J.M. Drake, P. López and J.C. Palencia, “Modeling distributed real-time systems with MAST 2,” Journal of Systems Architecture, vol 56, no. 6, Elsevier, pp. 331-340, 2013.

[3] Object Management Group, “UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems,” 2009 OMG Document, v1.0 formal/2009-11-02.

[4] J. Lehoczky, L. Sha and Y. Ding, "The rate monotonic scheduling algorithm: Exact characterization and average case behavior," In Real Time Systems Symposium (RTSS), pp. 166-171, 1989.

[5] E. Bini and G.C. Buttazzo, "Measuring the performance of schedulability tests," Real-Time Systems, vol 30, no. 1-2, pp. 129-154, 2005.

[6] K.W. Tindell and J. Clark, “Holistic Schedulability Analysis for Distributed Hard Real-Time Systems,” Microprocessing and Microprogramming, vol. 50, no. 2-3, pp. 117-134, 1994.

[7] K.W. Tindell, “Adding Time-Offsets to Schedulability Analysis,” Department of Computer Science, University of York, Technical Report YCS-221, 1994.

[8] J. Mäki-Turja and M. Nolin, “Efficient implementation of tight response-times for tasks with offsets,” Real-Time Systems Journal, vol. 40, no. 1, pp. 77-116, 2008.

[9] J.C. Palencia and M. González Harbour, “Exploiting Precedence Relations in the Schedulability Analysis of Distributed Real-Time Systems,” Proc. of the 20th Real-Time Systems Symposium (RTSS), pp. 328-339, 1999.

[10] J.M. Rivas, J.J. Gutiérrez, J.C. Palencia and M. González Harbour, “Deadline Assignment in EDF Schedulers for Real-Time Distributed Systems," In press, IEEE Transactions on Parallel and Distributed Systems.

[11] B. Kao and H. Garcia-Molina, "Deadline Assignment in a Distributed Soft Real-Time System," IEEE Transactions On Parallel And Distributed Systems (TPDS), vol. 8, no. 12 1997.

[12] J.W.S. Liu, "Real-time systems," Prentice Hall, 2000.

[13] J.M. Rivas, J.J. Gutiérrez, J.C. Palencia and M. González Harbour, "Schedulability analysis and optimization of heterogeneous EDF and FP distributed real-time systems," Proc. of the 23rd Euromicro Conference on Real-Time Systems (ECRTS), Porto (Portugal), pp. 195-204, 2011.