

# 1 DDMon — overview

(NOTE: this overview is also available in the artifact, in the file *OVERVIEW.md*.)

## 1.1 Introduction

DDMon is a deadlock monitoring tool for Erlang and Elixir programs based on the `gen_server` behaviour. We described DDMon in the *Section 7* and *Appendix A* of the companion paper.

This artifact supports the following claims and results from the paper:

- The results from *Section 7.2* (i.e., the plots in *Figures 15 and 16*) and the monitoring logs in *Appendix A.1* (*Listings 2, 3, and 4*) can be reproduced by following the instructions in *EVALUATION.md*.
- The claim of *Section 7.1* about DDMon’s applicability to `gen_server`-based systems for deadlock detection is documented in *EXAMPLE.md*.
- The test scenario DSL described in *Appendix A.1* (used to produce the plots in the paper) is documented in *SCENARIOS.md* with several examples.
  - One of such examples (called **routing**) is based on the encoding of replicated services discussed in *Appendix B* of the paper.

The *.md* files referenced above and below (and their rendering in PDF format) are included in the artifact.

NOTE: as explained in the Data Availability Statement in the paper, the mechanised proofs are not part of this artifact; they were provided as supplementary materials with the companion paper.

## 1.2 Hardware dependencies

To reproduce all the plots in the paper, the following minimal hardware prerequisites need to be met:

- 32GB of RAM
- 23GB of free disc space
- 8-core CPU (we have tested on 64-bit x86 and Apple M1)

We also provide smaller variants of the largest benchmark. For those, the following should suffice:

- 16GB of RAM
- 10GB of free disc space
- 4-core CPU

## 1.3 Getting Started Guide

We recommend using Docker for running our artifact. The file *EVALUATION.md* explains how to build the Docker image and how to perform the “kick the tires” assessment.

## 1.4 Step-by-Step Instructions

For reproducing the plots and logs in our paper, see the detailed instructions in *EVALUATION.md*, after the “kick the tires” section.

## 1.5 Reusability Guide

For evaluating the reusability of our artifact, we provide the following documentation.

- An example documenting how to use DDMon for monitoring an application based on the `gen_server` behaviour. The detailed instructions are available in *EXAMPLE.md*.
- Documentation of the test scenario DSL described in *Appendix A.1* of the companion paper (also used to produce the plots in the paper). The documentation is available in *SCENARIOS.md*, with several examples.
- Documentation about the internals of DDMon and its connection to the formalisation in the paper: see *IMPLEMENTATION.md*.
- Instructions for building and using DDMon without depending on Docker: see *README.md*.

Moreover, the file *BADGING.md* contains a more detailed overview of how we address the requirements for each ACM artifact badge.