BRUMM: A Case for Predictable Memory Reclamation (Artifact)

Viktor Reusch
Barkhausen Institut, Dresden, Germany

Michael Roitzsch
Barkhausen Institut, Dresden, Germany

Horst Schirmeier
TU Dresden, Germany

Abstract

This artifact contains all source code material that has been created for the paper: "BRUMM: A Case for Predictable Memory Reclamation". BRUMM is a mechanism for accounting for the predicted reclamation latency of user-space memory mappings. It enables the system operator to limit reclamation latency via configurable, per-process-group budgets. A prototype of BRUMM has been implemented in the L4Re microkernel-based operating system. The artifact includes the modified L4Re sources, the benchmark programs, and the evaluation scripts for generating plots. This artifact description provides details about the artifact and how to get started reproducing the paper's results.

2012 ACM Subject Classification Computer systems organization → Real-time operating systems; Security and privacy → Operating systems security; Security and privacy → Intrusion/anomaly detection and malware mitigation; Software and its engineering → Memory management; Computer systems organization → Availability

Keywords and phrases Resource Reclamation, Main Memory, Accounting, Operating System, Microkernel, Capability, L4Re

Digital Object Identifier 10.4230/DARTS.12.2.3

Related Article Viktor Reusch, Michael Roitzsch, and Horst Schirmeier, "BRUMM: A Case for Predictable Memory Reclamation", in 38th European Conference on Real-Time Systems (ECRTS 2026), LIPIcs, Vol. 375, pp. 15:1–15:24, 2026. `https://doi.org/10.4230/LIPIcs.ECRTS.2026.15`

Related Conference 38th European Conference on Real-Time Systems (ECRTS 2026), July 7–10, 2026, Lund, Sweden

1 Scope

This artifact contains all source code material of the BRUMM prototype, the benchmark programs, and the evaluation scripts of the accompanying paper. This artifact can be used to reproduce all the plots and measurements in the paper and to verify the claims made therein. However, the evaluation setup is automated only for a virtual machine environment; results may differ from those obtained with the physical machine setup used in the paper.

2 Content

The artifact package includes the following source code directories and files:

- `automation/`: Contains scripts and tools for automating the evaluation process.
- `l4re/`: Contains the L4Re operating system and related code used in the evaluation.
- `l4re/src/fiasco/`: Contains the L4Re microkernel source code.

© Viktor Reusch, Michael Roitzsch, and Horst Schirmeier; licensed under Creative Commons License CC-BY 4.0

- `l4re/src/fiasco/src/kern/reclam.cpp`: Is particularly interesting for the reclamation budgeting code of the BRUMM prototype.
- `l4re/src/l4/`: Contains the user-space source code.
- `l4re/src/l4/pkg/brumm/`: Contains the L4Re benchmarks used in the evaluation.
- `linux/`: Contains the Linux user-space benchmarks used in the evaluation.
- `measurements/`: Will contain the raw measurement data collected after running the benchmark scripts.
- `evaluation/`: Contains scripts for analyzing the measurement data and generating plots.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

4 Tested platforms

The artifact has been tested using a virtual environment on a single machine with the following specifications:

- x86_64 architecture
- At least 16 cores, preferably without hyperthreading and with homogeneous performance across cores
- At least 8 GiB of main memory available to the evaluation environment
- Linux
- KVM
- Bash
- One of:
  - nixpkgs with flakes
  - Docker
  - Podman
- `vm.max_map_count` should be set to at least 600 000

The paper originally performed all measurements bare-metal on a dual-socket server featuring two Intel Xeon Platinum 8358 CPUs (32 cores per CPU) and 512 GiB of main memory.

5 License

The artifact is made up of a variety of open source projects, which are each licensed under their respective licenses. Please refer to the respective `LICENSE` and `LICENSE.spdx` files in the projects' directories for details on the licenses of the individual components of the artifact.

6 MD5 sum of the artifact

85f5da1352fa0965bc7c7bf6fc520a7b

7 Size of the artifact

42.2 MiB

A Getting started with the artifact

To build the artifact's source code and run the included benchmarks, please follow the instructions in the `README.md` file in the root of the artifact package.