# Efficient NVRAM-based General-Purpose Operating Systems

Oliver Giersch[1], Henriette Hofmeier[3], Dustin Nguyen[2], Jonas Rabenstein[2], Christian Eichler[3]

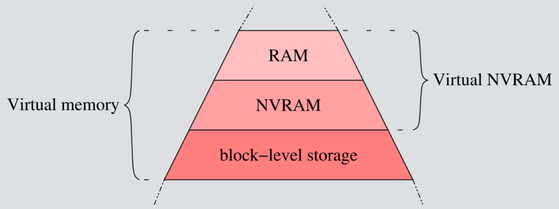
*[1]BTU Cottbus-Senftenberg [2]Friedrich-Alexander-Universität Erlangen-Nürnberg [3]Ruhr-Universität Bochum*

## NVRAM-based General-Purpose Operating Systems

- state-of-the-art operating systems, like Linux and FreeBSD
  → currently NVRAM-compatible but not NVRAM-exclusive
- operating-system kernel is moved entirely into NVRAM
  → inherently persistent operating-system execution

⇒ ***NVRAM-based Linux and FreeBSD***

## Integrating NVRAM into the Memory Hierarchy




⇒ ***functional transparency***

- integration via the virtual-memory subsystem
- two-level hierarchy of software-managed caches to hide memory-access latencies [4]:
  - NVRAM as a buffer for storage
  - DRAM as a buffer for NVRAM

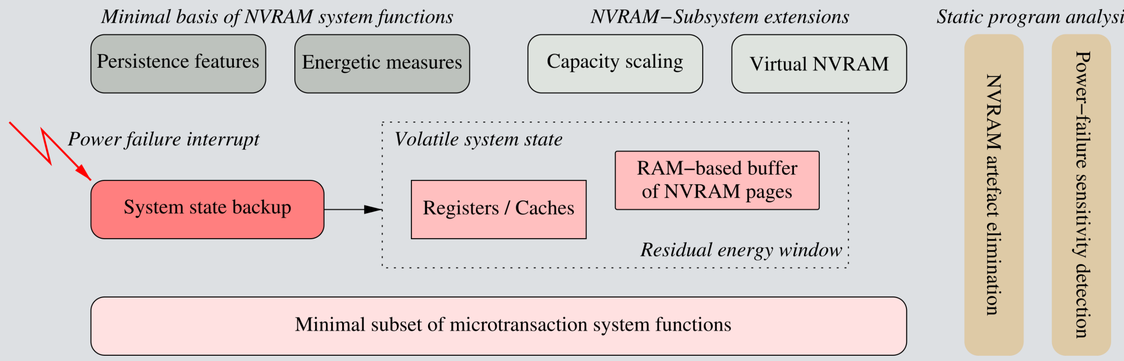
## NVRAM-based Operating-System Design

**Whole-System Persistence** [3]

- remaining volatile system state: CPU registers, caches and RAM buffer of NVRAM
- persisting volatile state with micro-checkpoints as opposed to transactional approaches (e.g., [2])
- utilizing static analysis to eliminate obsolete persistence measures

**Allocating with Persistence-Awareness**

- placing frequently accessed pages in DRAM
- utilizing DRAM as an NVRAM cache




⇒ ***software-based whole-system persistence***

**Ensuring Power-Failure Resistance**

- constant power monitoring by PSU
- *power failure interrupt* (PFI) initiates micro-checkpoint [1]
- handling power-failure sensitive sections [5], including NMI nesting and blocked IRQs

**Persisting Stateful Devices**

- recording device configurations at runtime
- persisting state to NVRAM on PFI
- *replay* on system recovery

## Potential Efficiency Improvements

- non-volatile design can compensate NVRAM's lower performance
  → removing traditional but now superfluous functionality, e.g., persistency measures, to increase system's efficiency
  → reduced Trusted Code Base (TCB)
- persistent main memory allows faster suspend-resume routines (*suspend-to-NVRAM*)
  → adapt dynamically to available, regenerative power supply with minimal overhead

⇒ ***improved carbon and energy efficiency through NVRAM-based OS design***

## References

[1] C. Eichler, H. Hofmeier, S. Reif, T. Hönig, J. Nolte, W. Schröder-Preikschat. Neverlast: A*n NVM-centric Operating System for Persistent Edge Systems*. Proc. APSys '21, ACM, 2021
[2] G. Heiser, E. Le Sueur, A. Danis, A. Budzynowski, T. I. Salomie, G. Alonso. *RapiLog: Reducing System Complexity through Verification*. Proc. EuroSys '13, ACM, 2013
[3] D. Narayana, O. Hodson. *Whole-System Persistence*. Proc. ASPLOS XVII, ACM, 2012
[4] I. B. Peng, M. B. Gokhale, E. W. Green. *System Evaluation of the Intel Optane Byte-addressable NVM*. Proc. MEMSYS '19, ACM, 2019
[5] B. Ransford, B. Lucia. *Nonvolatile Memory is a Broken Time Machine*. Proc. MSPC '14, ACM, 2014

Brandenburg University of Technology Cottbus - Senftenberg

Friedrich-Alexander-Universität Erlangen-Nürnberg

RUHR UNIVERSITÄT BOCHUM

Funded by Deutsche Forschungsgemeinschaft German Research Foundation
Project No. 465958100 501993201 502228341