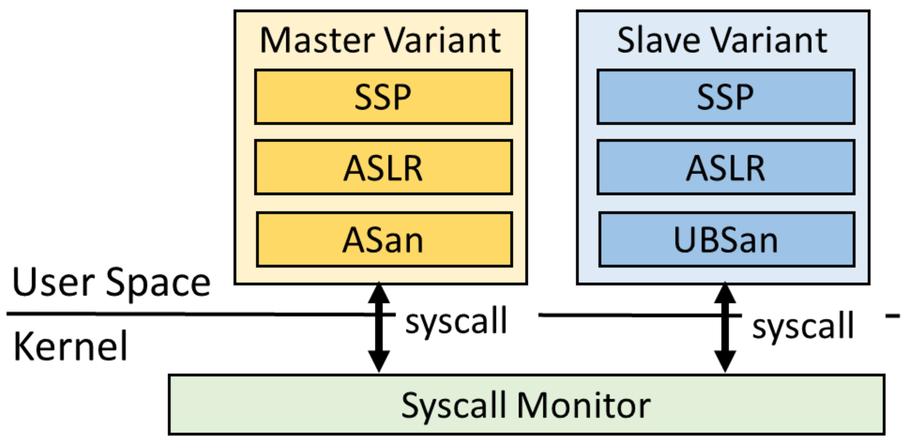
A Multi-variant Execution Environment for In-memory Databases

Shuhei Enomoto (Student)†, and Hiroshi Yamada†
†TUAT

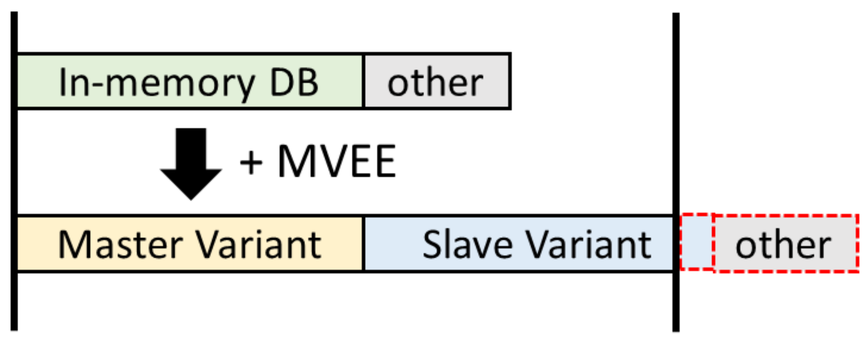
Software vulnerabilities

- Modern system software is still written by unsafe languages
- A number of security mechanism is supported
  - ASLR and SSP are can be bypassed with Information leaks
  - Sanitizers cannot defend different types of attacks
- The multi-variant execution environment (MVEE) is a promising approach



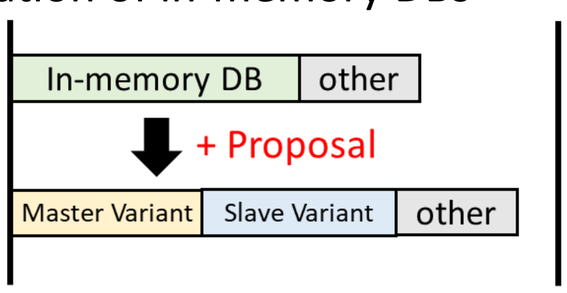
Problem

- In-memory databases are difficult to apply MVEE
  - Cause quite large memory space overhead
- In-memory databases also suffer from memory vulnerabilities
  - CVE-2019-10192, CVE-2019-10193 (Redis)
  - CVE-2019-15026, CVE-2019-11596 (Memcached)



Proposal

- MVEE runtime for in-memory databases
  - Reduces memory consumption
  - Enhances security as same as existing MVEE
  - No modification of in-memory DBs



Approach

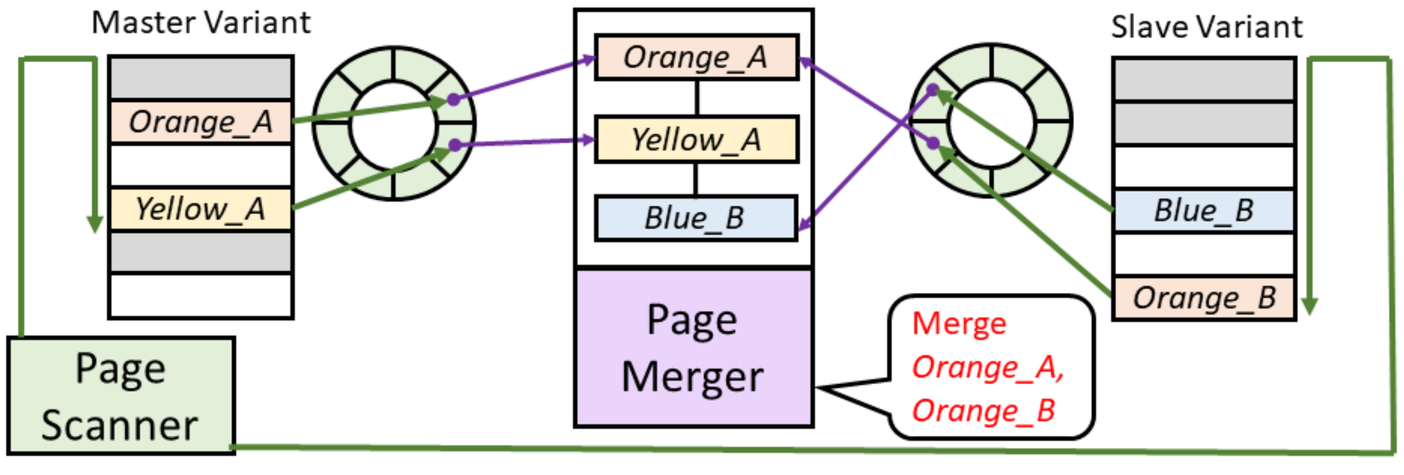
- Shares the memory contents among variants
  - Observation: In-memory DB variants have the similar memory contents to each other

Design

- Page Scanning
- Selects pages from address space of variants
  - Conditions (1): anonymous page
  - Conditions (2): stable content page

- Page Merging
- Merges same pages selected by scanner
  - Makes PTE pointed to merged page
    - Sets write protect flag
  - Releases other pages

- Syscall Monitoring
- Synchronizes syscalls
  - Each variant is given the same inputs



implementation

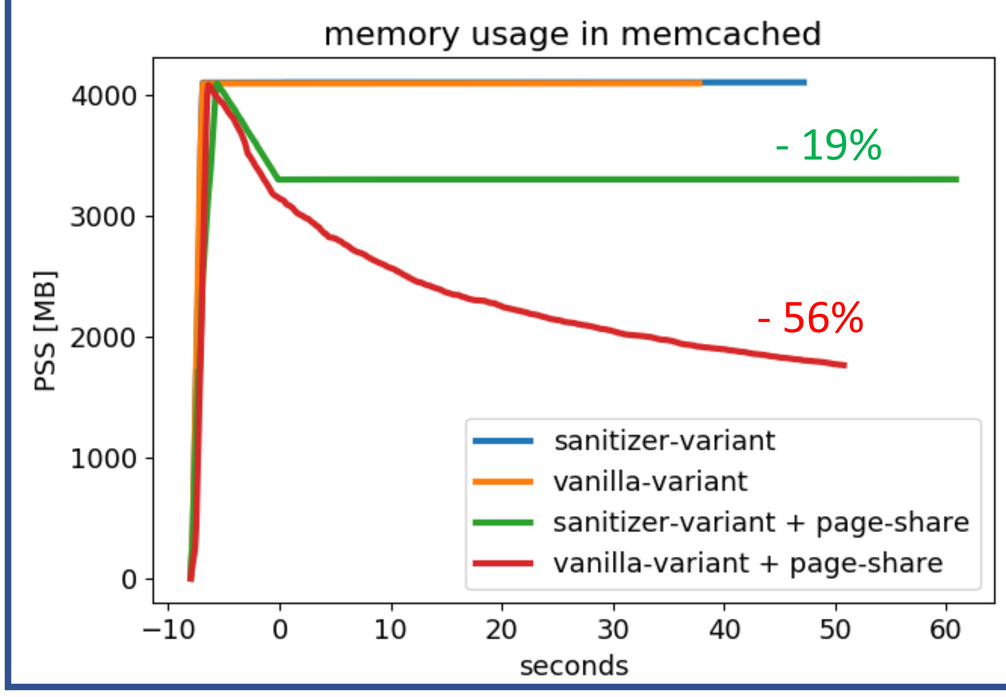
- Implemented into Linux 4.4.185
  - Page scanner and Merger: 411 LoC
  - Syscall Monitor: 1018 LoC

Configuration

- CPU: Intel Xeon Processor 4 cores
- Memory: 8GB of RAM
- Variants: 2
  - Pattern(1): ASLR, SSP × 1
    ASLR, SSP × 1
  - Pattern(2): ASLR, SSP, ASan × 1
    ASLR, SSP, UBSan × 1

Experiment: Memory Usage

- Launches Memcached as variant and Tests memtier_benchmark
- Measures total PSS of variants



Next Plans

- Makes page-sharing mechanism more efficient
  - Scanning with selected range
- Supports for more workloads
  - Make low overhead system even if write-based workloads
- Tests for more in-memory DBs
  - Tested: Memcached, Redis
  - Future: SQLite, VoltDB