

# VerMem: Versioned Memory using Multilevel-Cell NVRAM



Doe Hyun Yoon, Robert S. Schreiber, Jichuan Chang, Naveen Muralimanohar, Partha Ranganathan, Paolo Faraboschi  
Intelligent Infrastructure Lab (IIL), Hewlett-Packard Labs

## Motivation

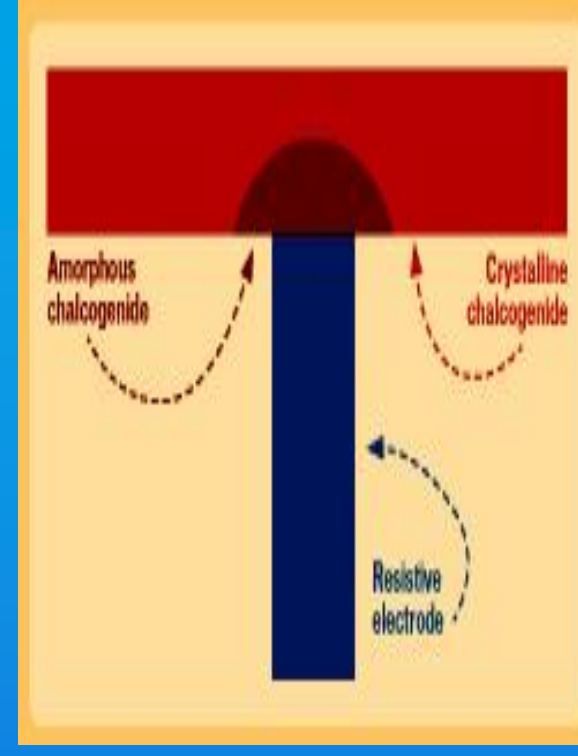
### Nonvolatile Byte-Addressable Memory

#### Opportunities:

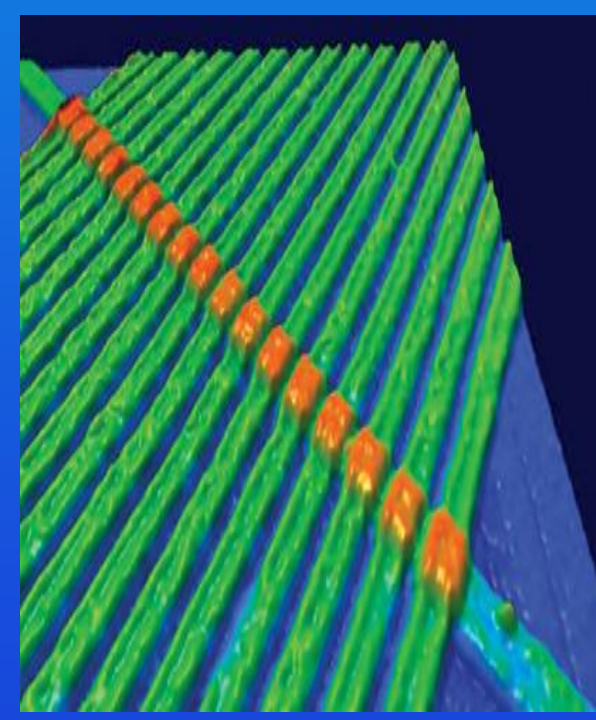
- Local in-memory checkpointing
- In-memory file system
  - + BPFS [Condit+ SOSP'09]
- Persistent data storage
  - + CDDS [Venkataraman+ FAST'11]
  - + NV-Heaps [Coburn+ ASPLOS'11]
  - + Mnemosyne [Volos+ ASPLOS'11]

#### Challenges:

- Atomic data updates
- Maintaining consistency



PCRAM



Memristor

### Multilevel-Cell NVRAM

Each cell stores several bits by a finer-grained quantization of the cell resistance



Unique asymmetry:

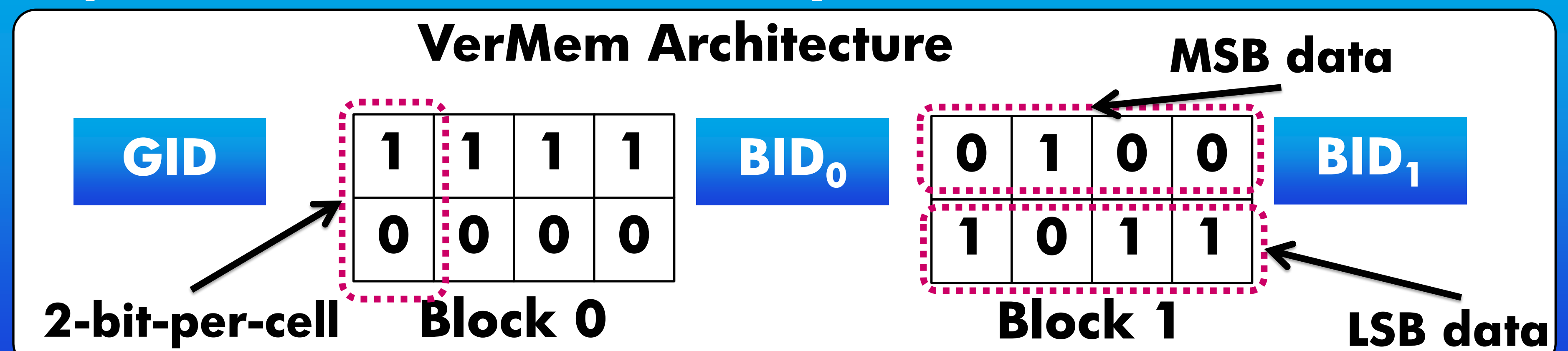
Reading MSB is faster than reading LSB

## VerMem

VerMem uses the MLC capability to implement a versioned memory

#### 64-bit counters

- GID: Global version number
- $BID_n$ : Per-Block version number for block n



## Version Management

Read fetches only MSB data

X Stable state

#### Initial State



#### Start Execution – Increment GID



#### Write '0101' to block 0 – read-modify-write



#### Write '1110' to block 0 – preserve LSB data

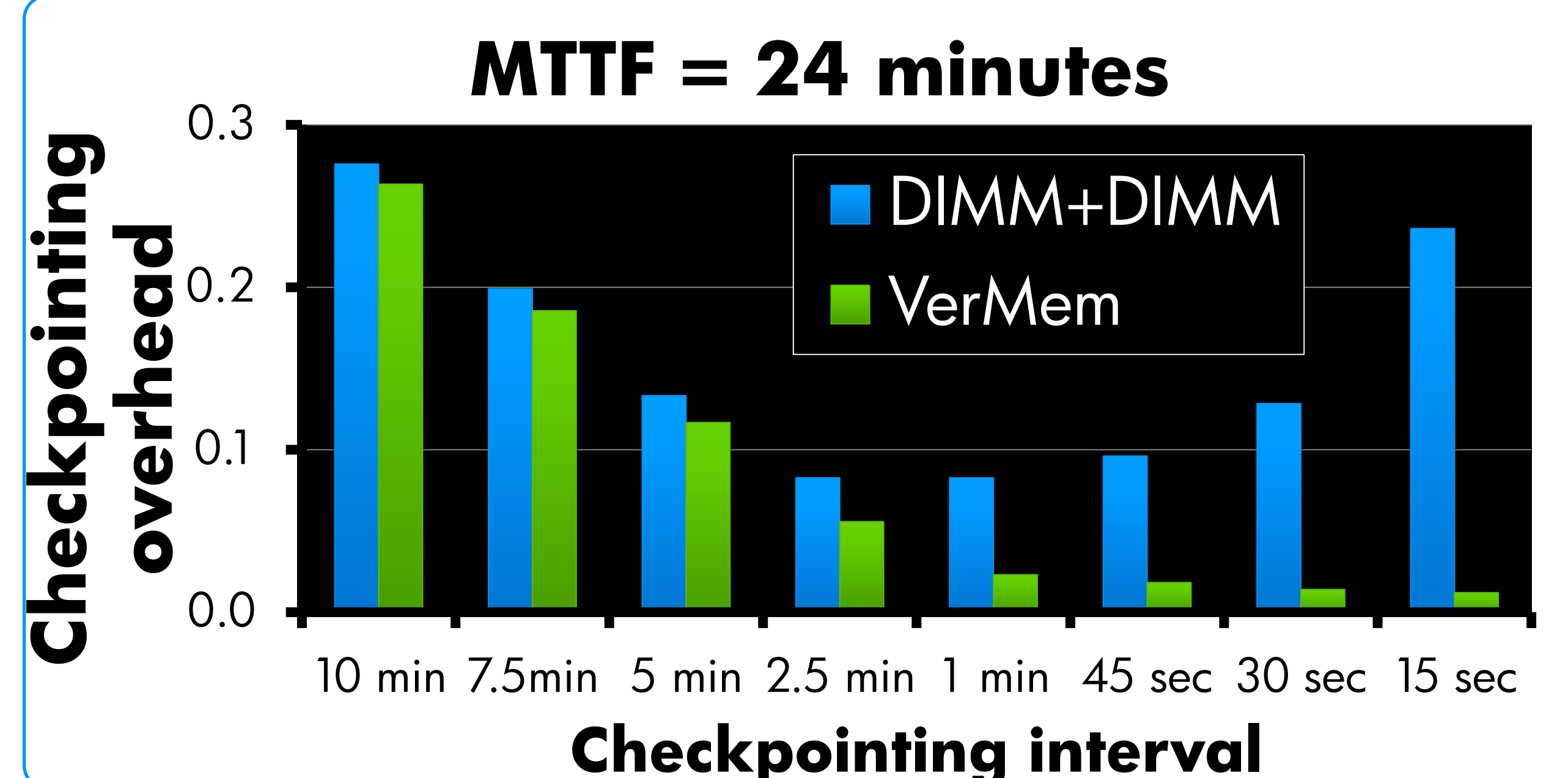
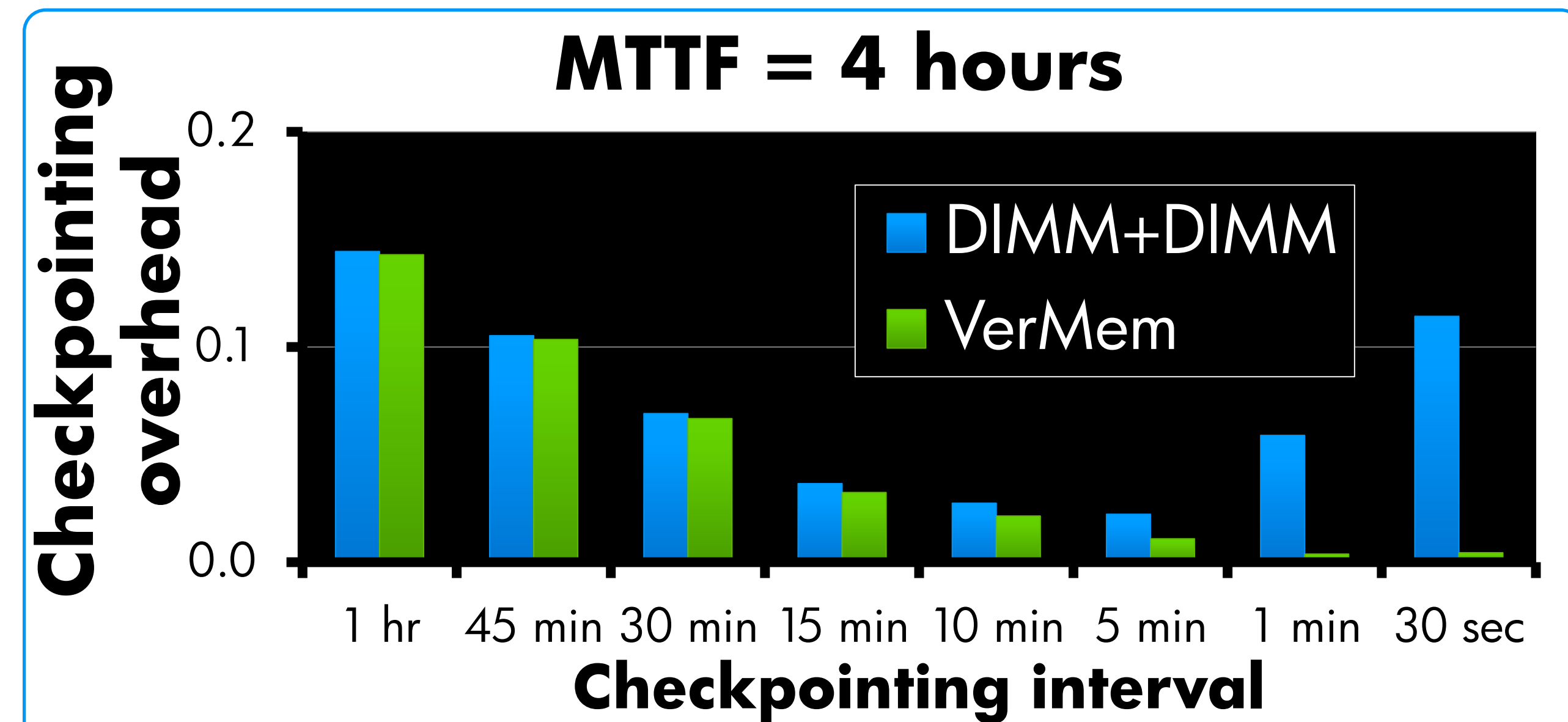


#### Checkpointing – just increment GID



CHECKPOINTING is a SINGLE-VARIABLE UPDATE !

## Evaluation



## Beyond Checkpointing

VerMem inherently supports ATOMIC and CONSISTENT data updates in NVRAM without logging

VerMem can be used as a substrate for in-memory persistent data objects