

Motivation

- Minimize explicit cache line write backs
- Checkpointing
- Build periodically-persistent data structures
- Directly modifying the data structure not feasible
- Reduce programming burden

Optane Characteristics

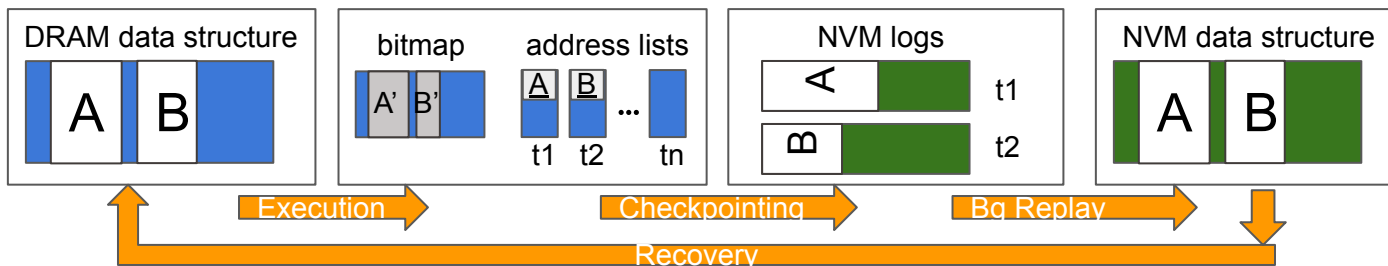


CpNvm

Idea: Write-combining + DRAM buffering

- 1) Program operates normally
- 2) CpNvm keeps track of modifications during execution
 - a) using bitmap + address lists
- 3) Checkpointing: append modifications to the end of the redo-log
- 4) Replay and recover from NVM replica [DudeTM, ASPLOS 2017]

CpNvm Design and Implementation



API

```
void* cpnvmInit(int bg_thrs=2, ...);
void cpnvmThreadInit();
void* cpnvmRoot(size_t size, ...);
void* cpnvmAlloc(size_t size, ...);
void* cpnvmFree(void* ptr, ...);
void* cpnvmMark(void* ptr, ...);
void* cpnvmCheckpoint();
bool cpnvmInRecovery();
```

Changes to Masstree, Memcached < 30 loC

Evaluation

