Attacking and defending PCM-based main memory

Milo Polte and Robert J. Simmons

- DRAM is running into density scaling problems
- Phase-Change Memory (PCM) is a NVRAM technology that uses thermal expansion properties to store data
- Previous work: PCM feasible as DRAM replacement
- Like Flash, PCM is susceptible to wearout.
    - **Attacking**: Worst case wear? Realistic wear?
    - **Defending**: Can it be mitigated with wear-leveling?

Attacking and defending PCM-based main memory

Milo Polte and Robert J. Simmons

- DRAM is running into density scaling problems
- Phase-Change Memory (PCM) is a NVRAM technology that uses thermal expansion properties to store data
- Previous work: PCM feasible as DRAM replacement
- Like Flash, PCM is susceptible to wearout.
  - **Attacking**: Worst case wear? Realistic wear?
  - **Defending**: Can it be mitigated with wear-leveling?

Attacking and defending PCM-based main memory

Milo Polte and Robert J. Simmons

- DRAM is running into density scaling problems
- Phase-Change Memory (PCM) is a NVRAM technology that uses thermal expansion properties to store data
- Previous work: PCM feasible as DRAM replacement
- Like Flash, PCM is susceptible to wearout.
    - **Attacking**: Worst case wear? Realistic wear?
    - **Defending**: Can it be mitigated with wear-leveling?

Attacking and defending PCM-based main memory

Milo Polte and Robert J. Simmons

- DRAM is running into density scaling problems
- Phase-Change Memory (PCM) is a NVRAM technology that uses thermal expansion properties to store data
- Previous work: PCM feasible as DRAM replacement
- Like Flash, PCM is susceptible to wearout.
    - **Attacking**: Worst case wear? Realistic wear?
    - **Defending**: Can it be mitigated with wear-leveling?

Attacking and defending PCM-based main memory

Milo Polte and Robert J. Simmons

- DRAM is running into density scaling problems
- Phase-Change Memory (PCM) is a NVRAM technology that uses thermal expansion properties to store data
- Previous work: PCM feasible as DRAM replacement
- Like Flash, PCM is susceptible to wearout.
  - **Attacking**: Worst case wear? Realistic wear?
  - **Defending**: Can it be mitigated with wear-leveling?

Attacking and defending PCM-based main memory

Milo Polte and Robert J. Simmons

- DRAM is running into density scaling problems
- Phase-Change Memory (PCM) is a NVRAM technology that uses thermal expansion properties to store data
- Previous work: PCM feasible as DRAM replacement
- Like Flash, PCM is susceptible to wearout.
    - **Attacking**: Worst case wear? Realistic wear?
    - **Defending**: Can it be mitigated with wear-leveling?

Previous work

Primary background: Lee et al., Architecting Phase Change Memory as a Scalable DRAM Alternative, ISCA 2009

- PCM is competitive with DRAM in terms of time and power
- Instead of one 2048-byte memory buffer (sense amplifier), have four 512-byte memory buffers
  - Mitigates comparatively slow writes
- Instead of writing back the whole buffer back to memory, track modified data by L2 cache blocks or by word
  - Prevents wearout

Previous work

Primary background: Lee et al., Architecting Phase Change Memory as a Scalable DRAM Alternative, ISCA 2009

- PCM is competitive with DRAM in terms of time and power
- Instead of one 2048-byte memory buffer (sense amplifier), have four 512-byte memory buffers
  - Mitigates comparatively slow writes
- Instead of writing back the whole buffer back to memory, track modified data by L2 cache blocks or by word
  - Prevents wearout

Previous work

Primary background: Lee et al., Architecting Phase Change Memory as a Scalable DRAM Alternative, ISCA 2009

- PCM is competitive with DRAM in terms of time and power
- Instead of one 2048-byte memory buffer (sense amplifier), have four 512-byte memory buffers
  - Mitigates comparatively slow writes
- Instead of writing back the whole buffer back to memory, track modified data by L2 cache blocks or by word
  - Prevents wearout

Previous work

Primary background: Lee et al., Architecting Phase Change Memory as a Scalable DRAM Alternative, ISCA 2009

- PCM is competitive with DRAM in terms of time and power
- Instead of one 2048-byte memory buffer (sense amplifier), have four 512-byte memory buffers
  - Mitigates comparatively slow writes
- Instead of writing back the whole buffer back to memory, track modified data by L2 cache blocks or by word
  - Prevents wearout

Experimental setup

Simulation Environment

- Components: Main memory, wear levelers, attackers

Wear leveler

- Intercepts requests to memory
- Can redirect reads and writes made to logical memory (size N) to physical memory (size N$\times\alpha$)

Attacker

- Generates requests to memory
- Degenerate requests
  - Always write to same logical location
  - Always write to same same physical location (if possible)
- Real requests derived from SPEC benchmarks
  - Turns out they can look pretty degenerate!

Experimental setup

Simulation Environment

- Components: Main memory, wear levelers, attackers

Wear leveler

- Intercepts requests to memory
- Can redirect reads and writes made to logical memory (size N) to physical memory (size N$\times\alpha$)

Attacker

- Generates requests to memory
- Degenerate requests
    - Always write to same logical location
    - Always write to same same physical location (if possible)
- Real requests derived from SPEC benchmarks
    - Turns out they can look pretty degenerate!

Experimental setup

Simulation Environment

- Components: Main memory, wear levelers, attackers

Wear leveler

- Intercepts requests to memory
- Can redirect reads and writes made to logical memory (size N) to physical memory (size N$\times\alpha$)

Attacker

- Generates requests to memory
- Degenerate requests
    - Always write to same logical location
    - Always write to same same physical location (if possible)
- Real requests derived from SPEC benchmarks
    - Turns out they can look pretty degenerate!

Experimental setup

Simulation Environment

- Components: Main memory, wear levelers, attackers

Wear leveler

- Intercepts requests to memory
- Can redirect reads and writes made to logical memory (size N) to physical memory (size N$\times\alpha$)

Attacker

- Generates requests to memory
- Degenerate requests
  - Always write to same logical location
  - Always write to same same physical location (if possible)
- Real requests derived from SPEC benchmarks
  - Turns out they can look pretty degenerate!

Experimental setup

Simulation Environment

- Components: Main memory, wear levelers, attackers

Wear leveler

- Intercepts requests to memory
- Can redirect reads and writes made to logical memory (size N) to physical memory (size N$\times\alpha$)

Attacker

- Generates requests to memory
- Degenerate requests
  - Always write to same logical location
  - Always write to same same physical location (if possible)
- Real requests derived from SPEC benchmarks
  - Turns out they can look pretty degenerate!

Experimental setup

Simulation Environment

- Components: Main memory, wear levelers, attackers

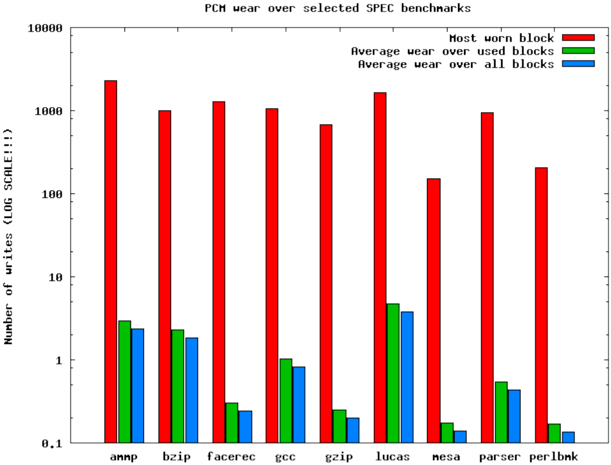
Wear leveler

- Intercepts requests to memory
- Can redirect reads and writes made to logical memory (size N) to physical memory (size N$\times\alpha$)

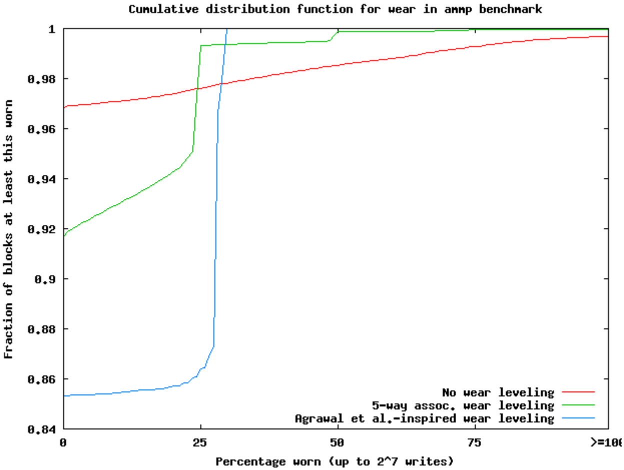
Attacker

- Generates requests to memory
- Degenerate requests
  - Always write to same logical location
  - Always write to same same physical location (if possible)
- Real requests derived from SPEC benchmarks
  - Turns out they can look pretty degenerate!

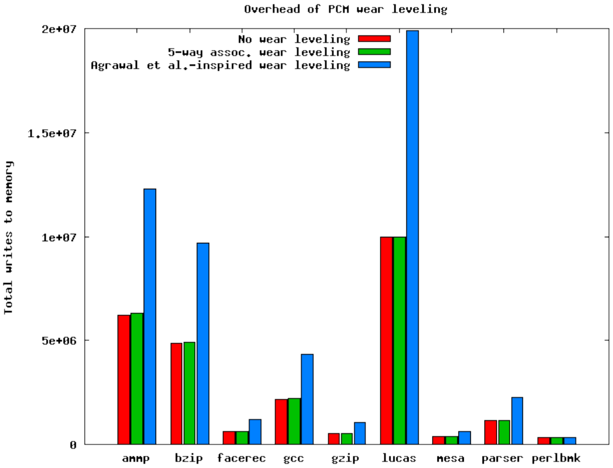
Necessity of wear leveling



Effect of wear leveling



Overhead of wear leveling



Conclusions and future work

Conclusions

- Relatively straight-forward traces can be surprisingly bad
- Relatively simple wear leveling can be surprisingly effective

Short-term future:

- Tweak existing algorithms
- Reduce overhead

Longer-term future:

- Evaluate timing
- Integrate with the existing MSR simulation infrastructure
- Different approaches: OS? “Write Victim Cache?”

Conclusions and future work

Conclusions

- Relatively straight-forward traces can be surprisingly bad
- Relatively simple wear leveling can be surprisingly effective

Short-term future:

- Tweak existing algorithms
- Reduce overhead

Longer-term future:

- Evaluate timing
- Integrate with the existing MSR simulation infrastructure
- Different approaches: OS? “Write Victim Cache?”

Conclusions and future work

Conclusions

- Relatively straight-forward traces can be surprisingly bad
- Relatively simple wear leveling can be surprisingly effective

Short-term future:

- Tweak existing algorithms
- Reduce overhead

Longer-term future:

- Evaluate timing
- Integrate with the existing MSR simulation infrastructure
- Different approaches: OS? “Write Victim Cache?”

Conclusions and future work

Conclusions

- Relatively straight-forward traces can be surprisingly bad
- Relatively simple wear leveling can be surprisingly effective

Short-term future:

- Tweak existing algorithms
- Reduce overhead

Longer-term future:

- Evaluate timing
- Integrate with the existing MSR simulation infrastructure
- Different approaches: OS? “Write Victim Cache?”