Improving DRAM Performance via Variable Refresh Latency

VRL-DRAM

Anup Das

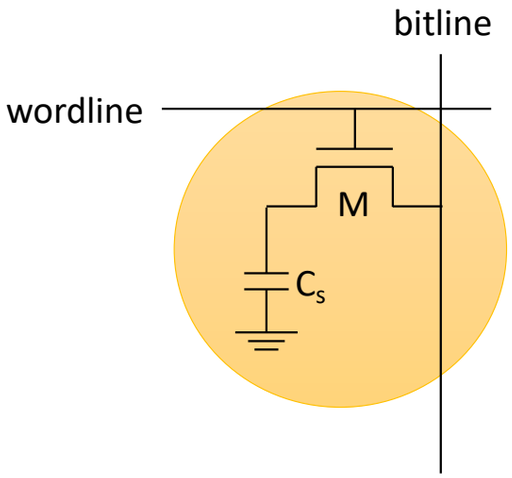
Hasan Hassan and Onur Mutlu

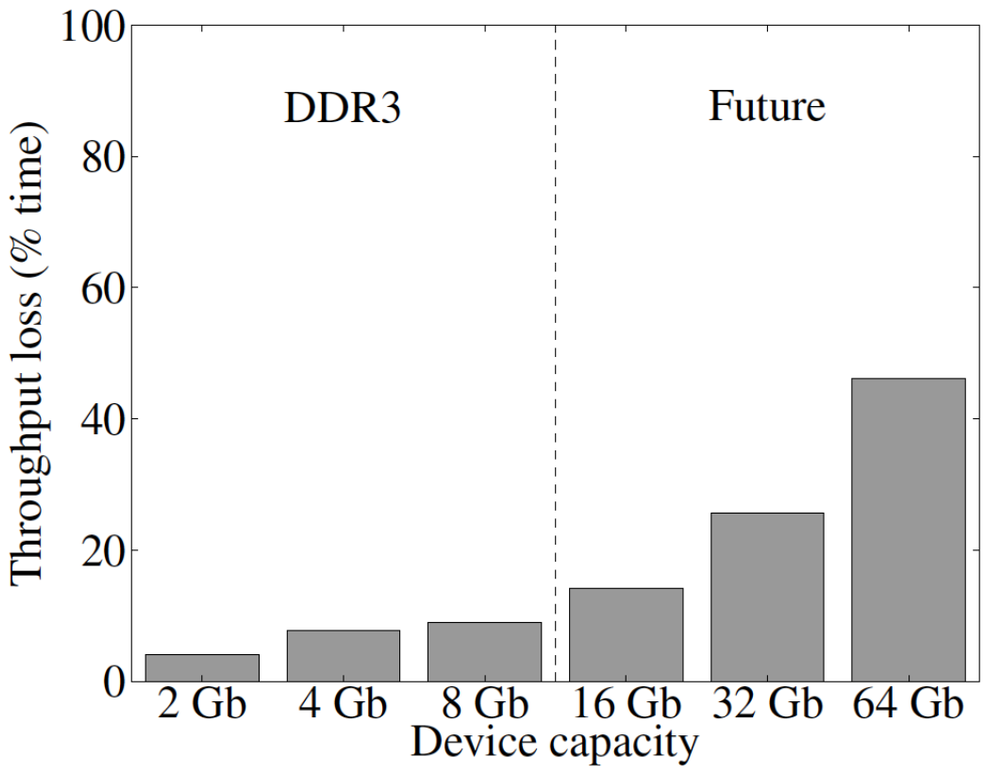
Executive Summary

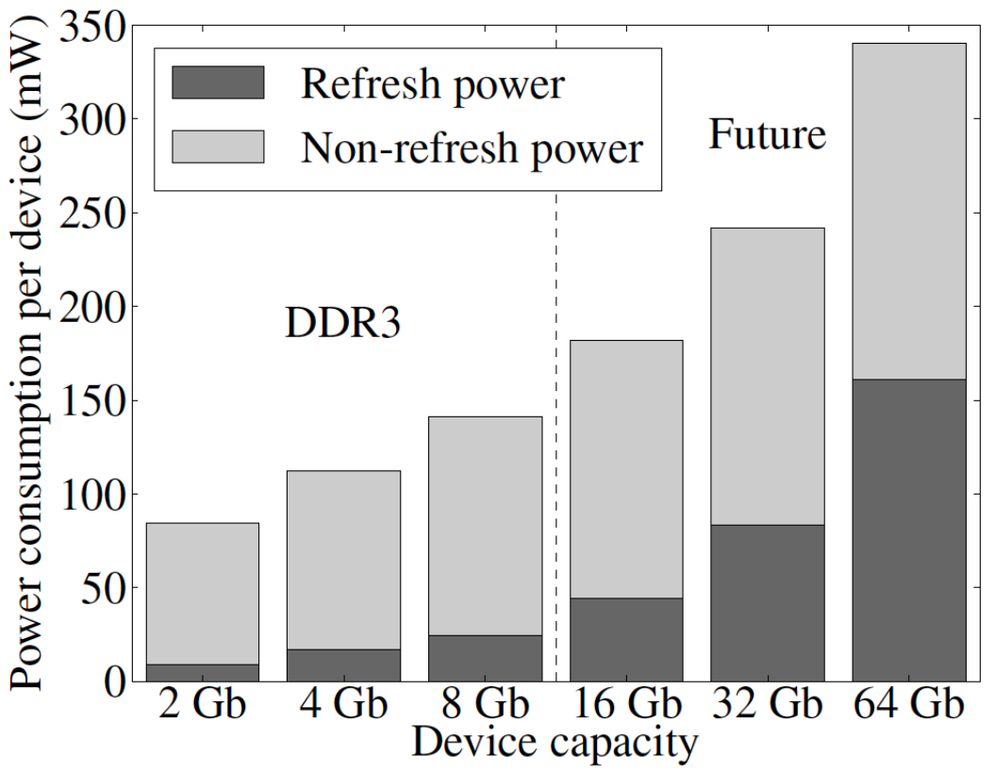
- Observations
  - During refresh, *almost half* of the refresh time is spent in injecting the *last 5% of charge* of a fully charged cell
  - Once *fully* restored, a DRAM cell can sustain multiple *partial* refreshes *without* sacrificing data integrity
- Idea: Variable Refresh Latency DRAM
  - Fully refresh a DRAM cell only when necessary, and otherwise issue partial refresh → reduces refresh latency in common case
- Characterization: new detailed circuit-level analytical model
  - Great potential to lower DRAM refresh timing parameters
- Performance Evaluation
  - Significant refresh overhead reduction (23% for PARSEC-3.0 workloads without errors)

DRAM Refresh Overhead

- DRAM cell loses charge over time
- Needs periodic refresh: once every 64ms
- Power overhead and performance loss due to DRAM refresh







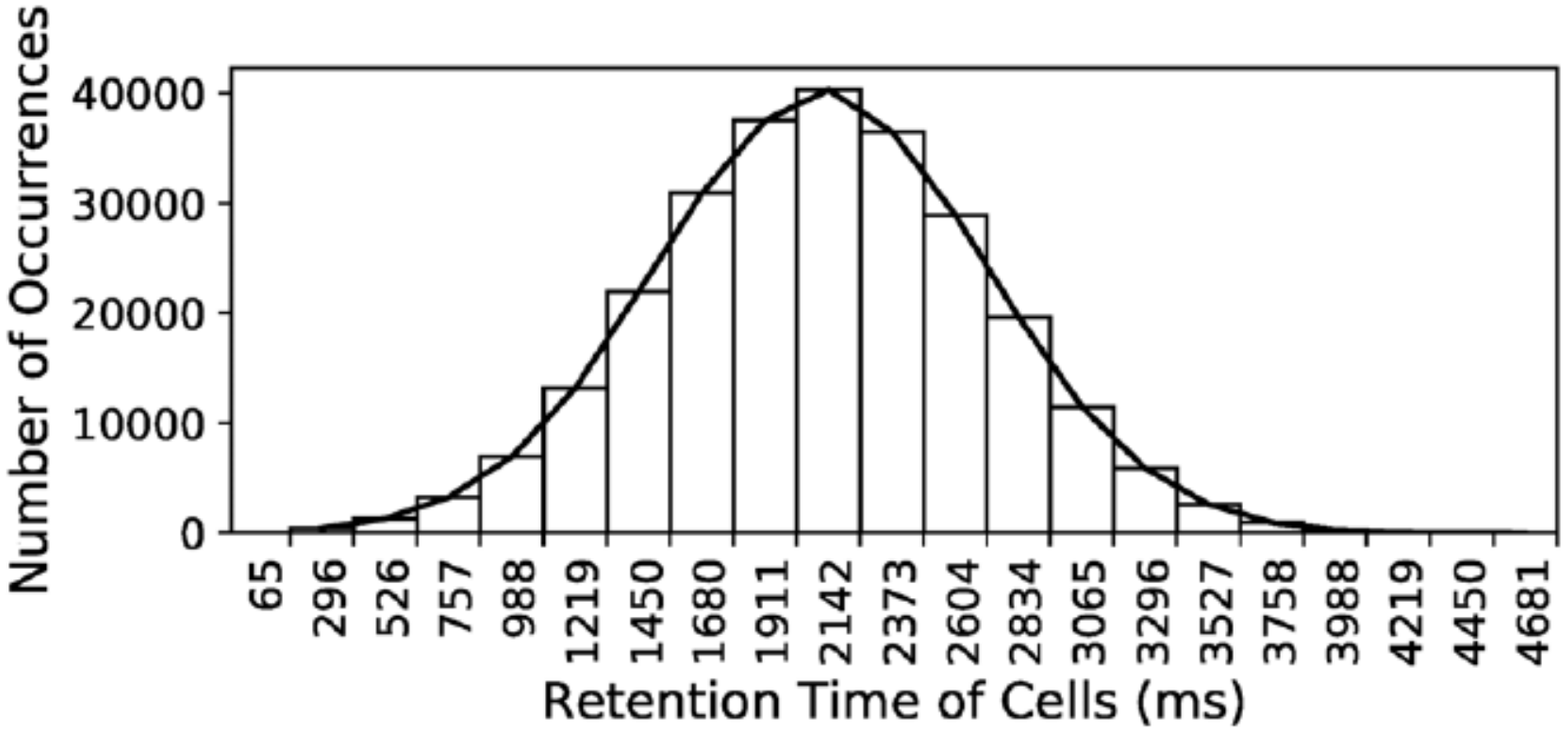
J. Liu et al., RAIDR: Retention-Aware Intelligent DRAM Refresh in ISCA'12

Mitigating DRAM Refresh Overhead

- Not all DRAM cells require the default 64ms refresh rate

- DRAM refresh overhead can be minimized by skipping refresh for cells that can retain data longer than 64ms

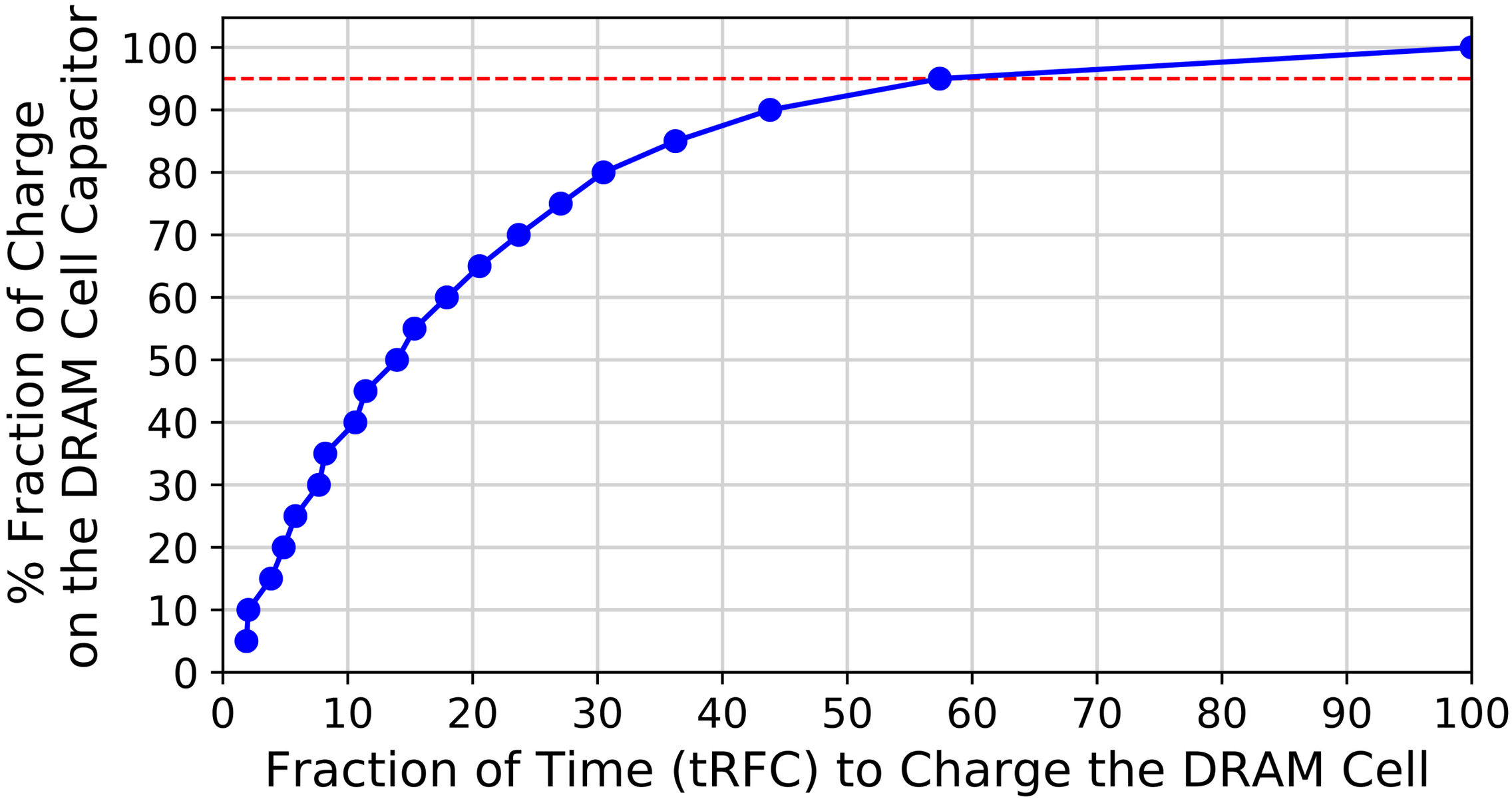
  - Liu et al., RAIDR, ISCA'12



Refresh period (ms)	Number of rows in a bank
64	68
128	101
192	145
256	7878

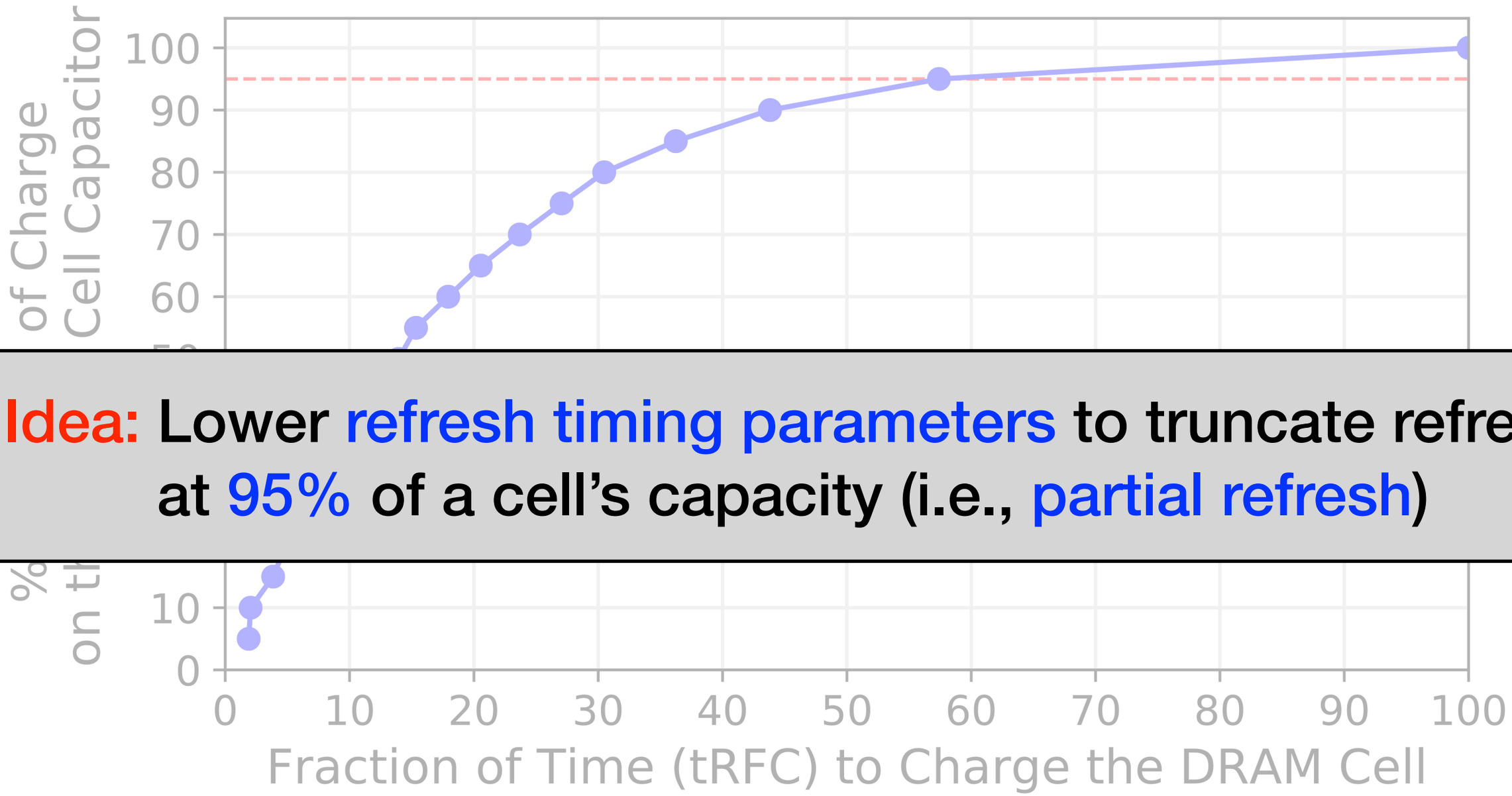
DRAM Refresh Related Key Observations (I)

- During refresh, almost half of the refresh time is spent in injecting the last 5% of charge of a fully charged cell



DRAM Refresh Related Key Observations (I)

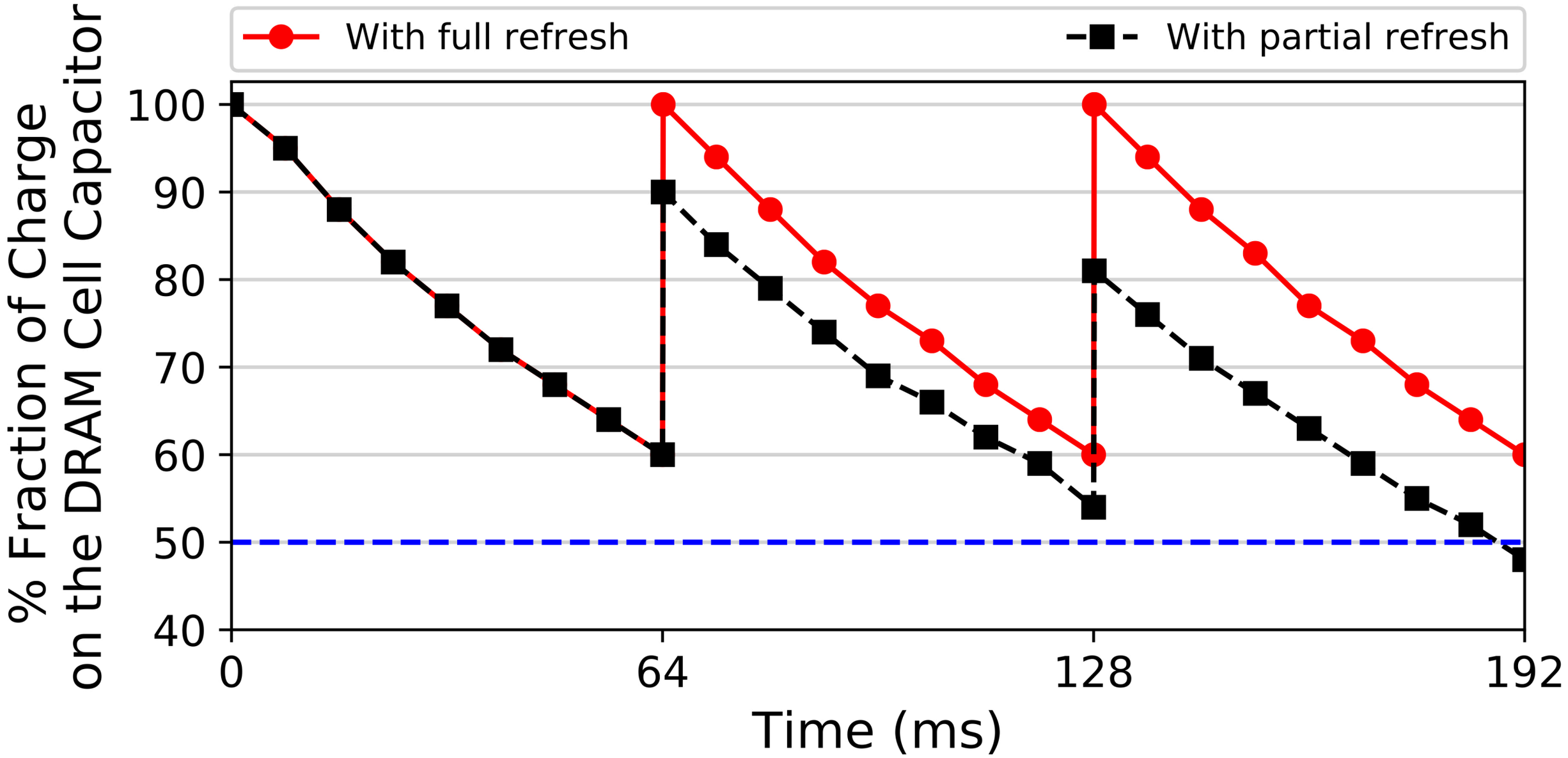
- During refresh, almost half of the refresh time is spent in injecting the last 5% of charge of a fully charged cell



Idea: Lower refresh timing parameters to truncate refresh at 95% of a cell's capacity (i.e., partial refresh)

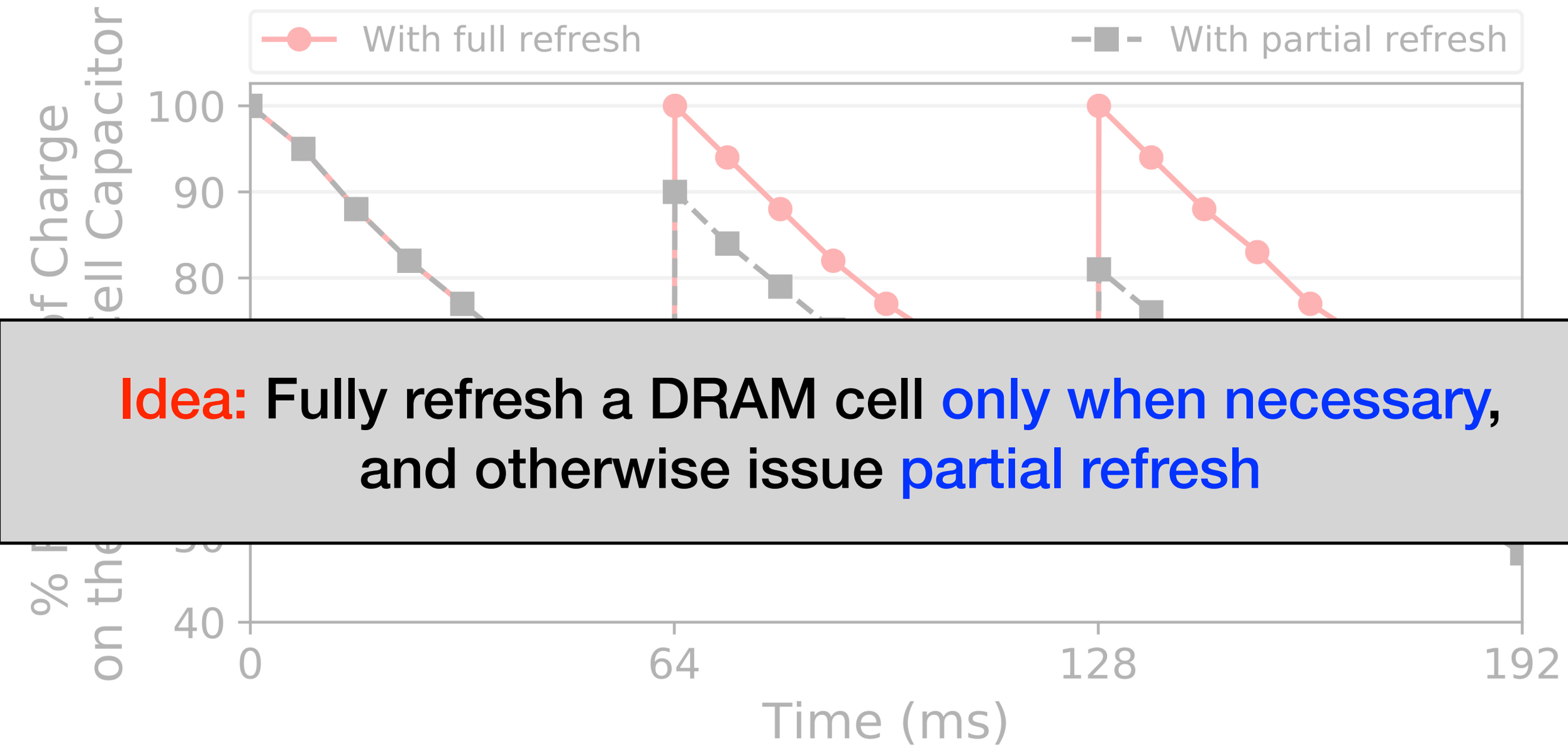
DRAM Refresh Related Key Observations (II)

- Once fully restored, a DRAM cell can sustain *multiple* partial refreshes without sacrificing data integrity



DRAM Refresh Related Key Observations (II)

- Once fully restored, a DRAM cell can sustain multiple partial refreshes without sacrificing data integrity



Idea: Fully refresh a DRAM cell only when necessary, and otherwise issue partial refresh

Variable Refresh Latency DRAM

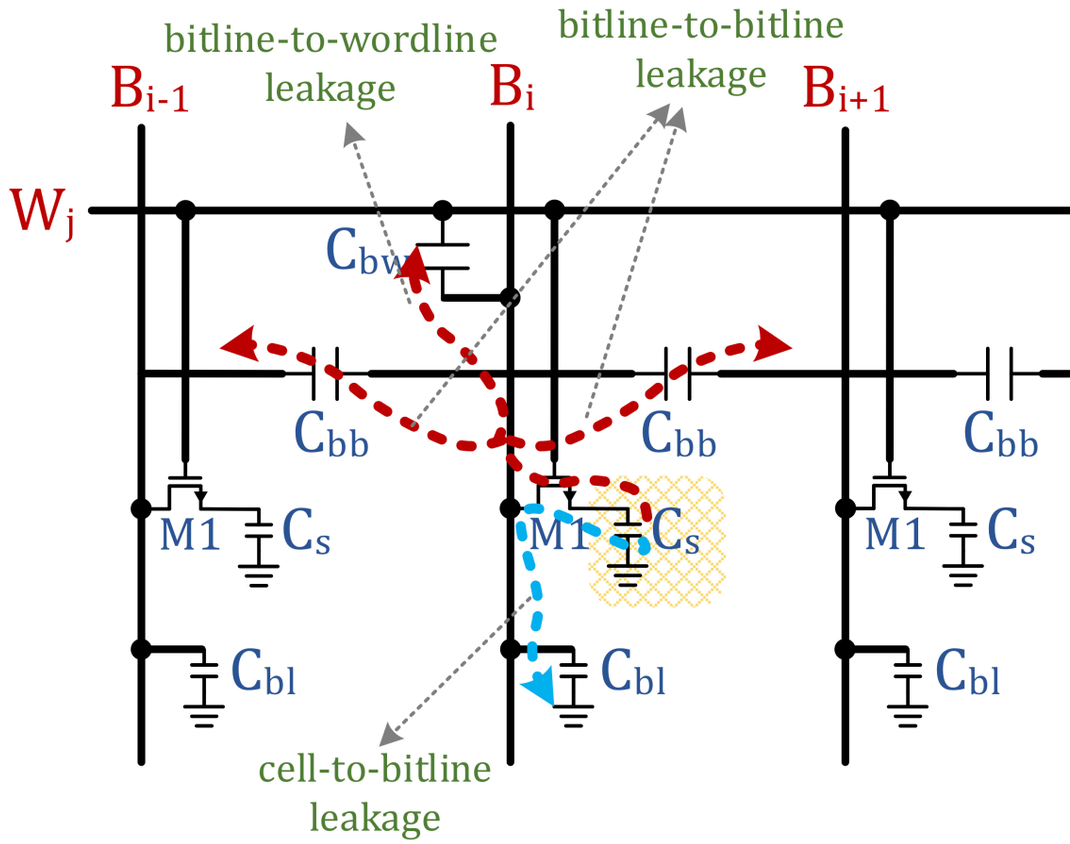
- Key Idea
  - Use two timing parameters for DRAM refresh
    - Full (slow) refresh: Issue only when necessary
    - Partial (fast) refresh: Issue when no problem with correctness
- Two components
  - Characterization
    - Accurately estimate the number of partial refreshes that a DRAM cell can reliably sustain
  - Scheduling
    - Whenever possible, issue partial refreshes to reduce the refresh overhead

Outline

- Introduction
- DRAM refresh characterization
- Partial and full refresh scheduling
- Evaluation
- Conclusion

DRAM Refresh Characterization (I)

- Existing DRAM circuit simulators do not take into account
  - Data pattern stored in DRAM
  - Sneak paths
  - Bitline/wordline parasitics



- SPICE simulation is time consuming

DRAM Refresh Characterization (I)

- Estimate number of partial refreshes that can be reliably sustained
  - A new detailed circuit-level analytical model for DRAM
    - Details in paper, also available as an open-source tool *https://github.com/anupkdas-nus/VRL-DRAM*
- End result:

Row ID	Number of partial refreshes that can be reliably sustained
0	3
1	1
. . .	. . .
127	5

DRAM Refresh Characterization (II)

- Characterize retention time of DRAM rows using Liu et al., RAIDR, ISCA'12
- End result:

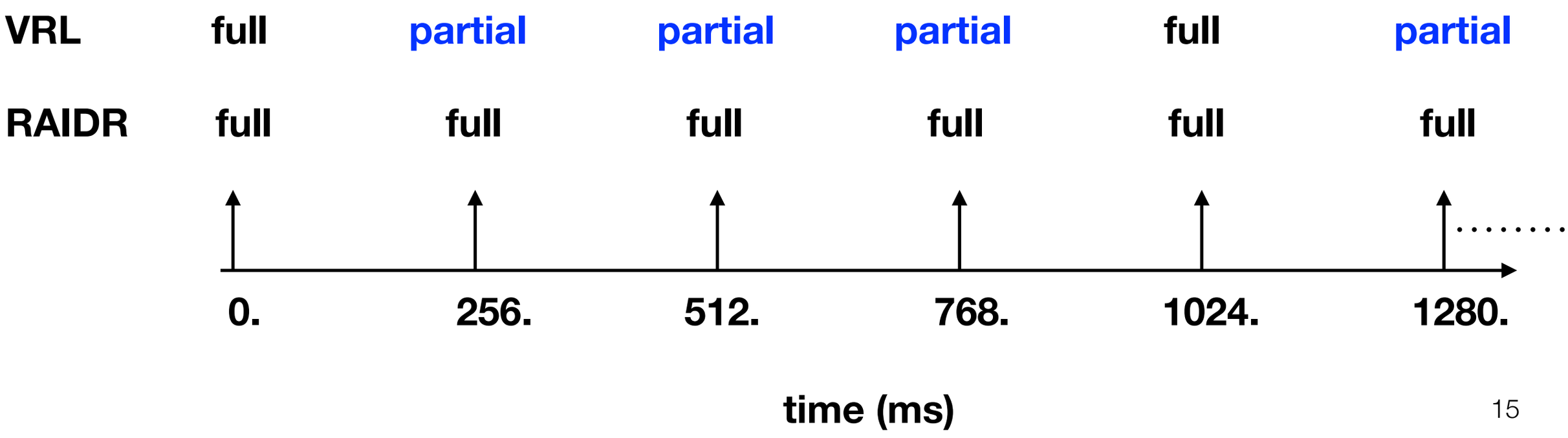
Row ID	Retention time
0	256
1	64
...	...
127	128

Outline

- Introduction
- DRAM refresh characterization
- Partial and full refresh scheduling
- Evaluation
- Conclusion

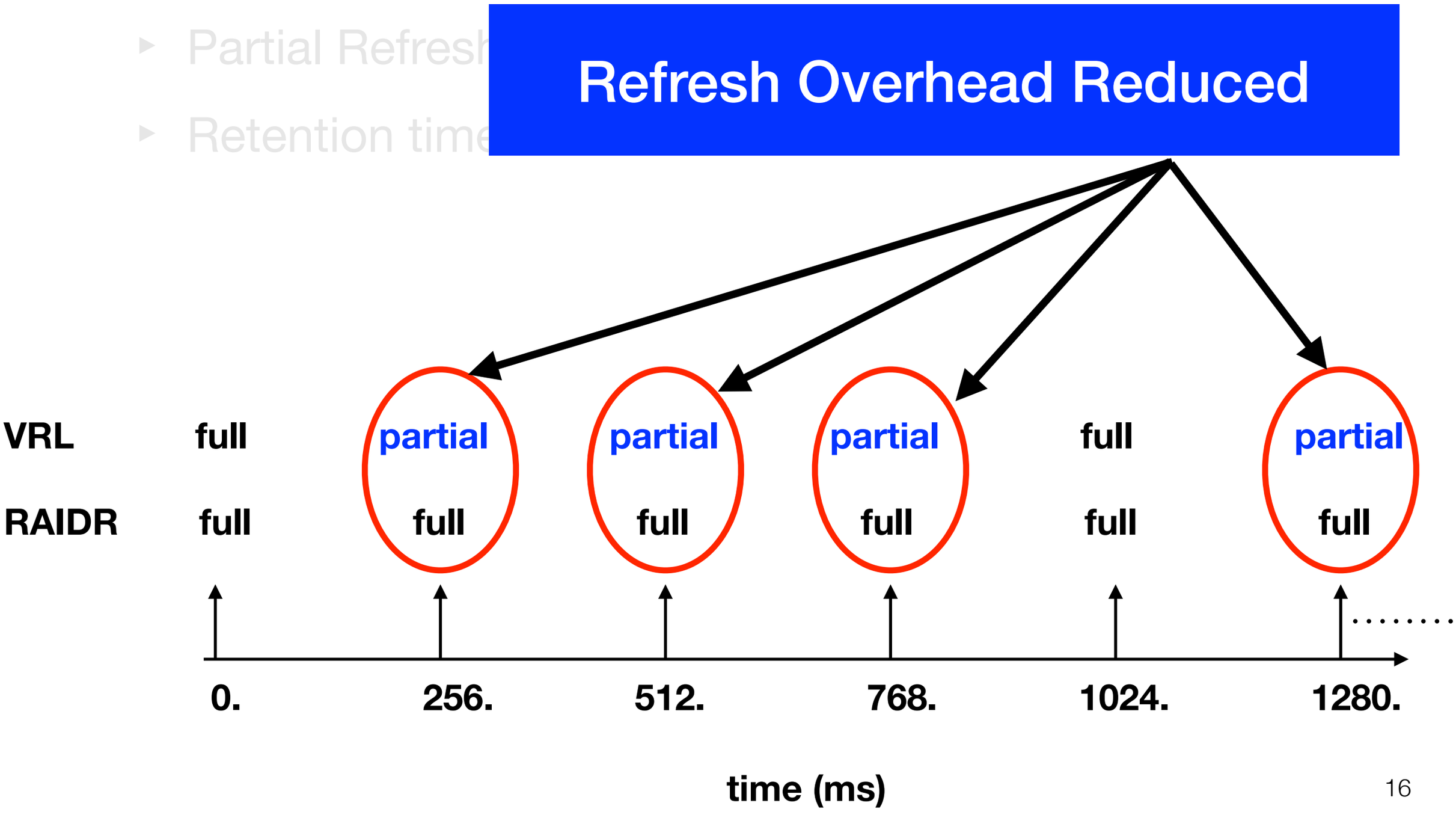
Partial Refresh Scheduling (I)

- Key Idea (I): ***VRL***
  - Row 0
    - Partial Refresh Interval = 3
    - Retention time = 256



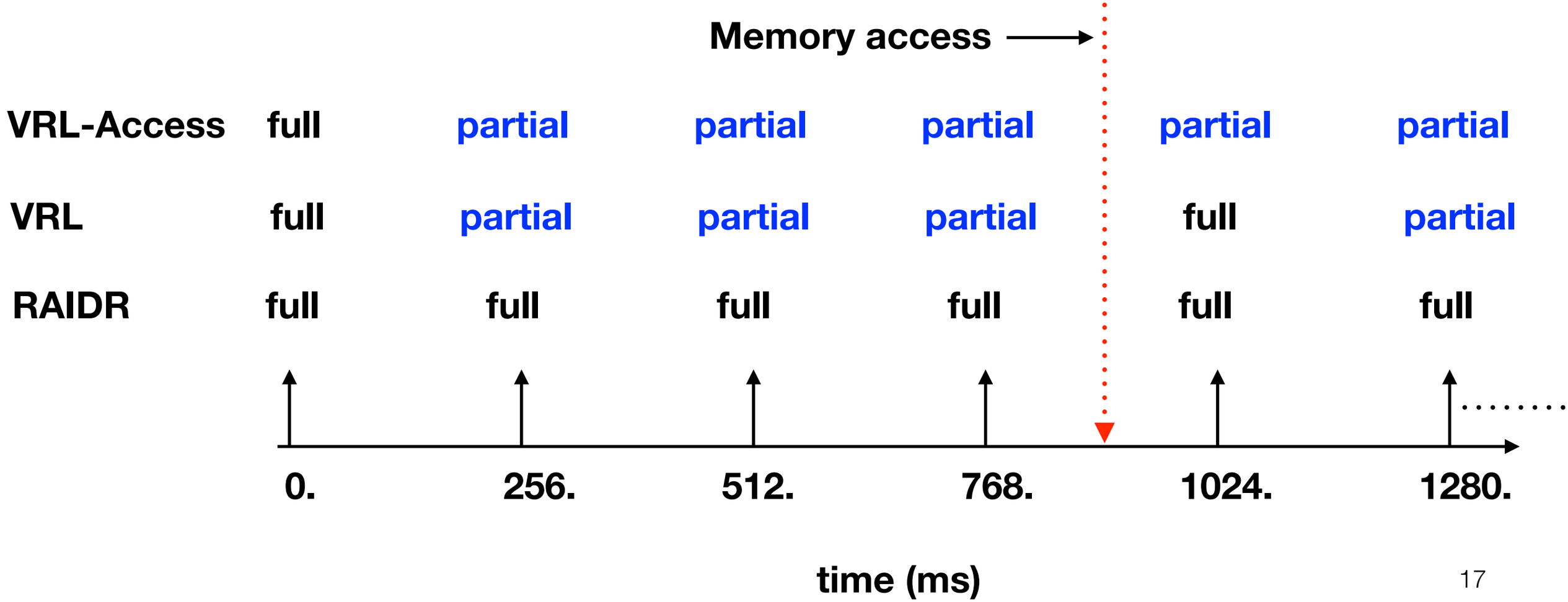
Partial Refresh Scheduling (I)

- Key Idea (1): ***VRL***
  - Row 0
    - Partial Refresh
    - Retention time



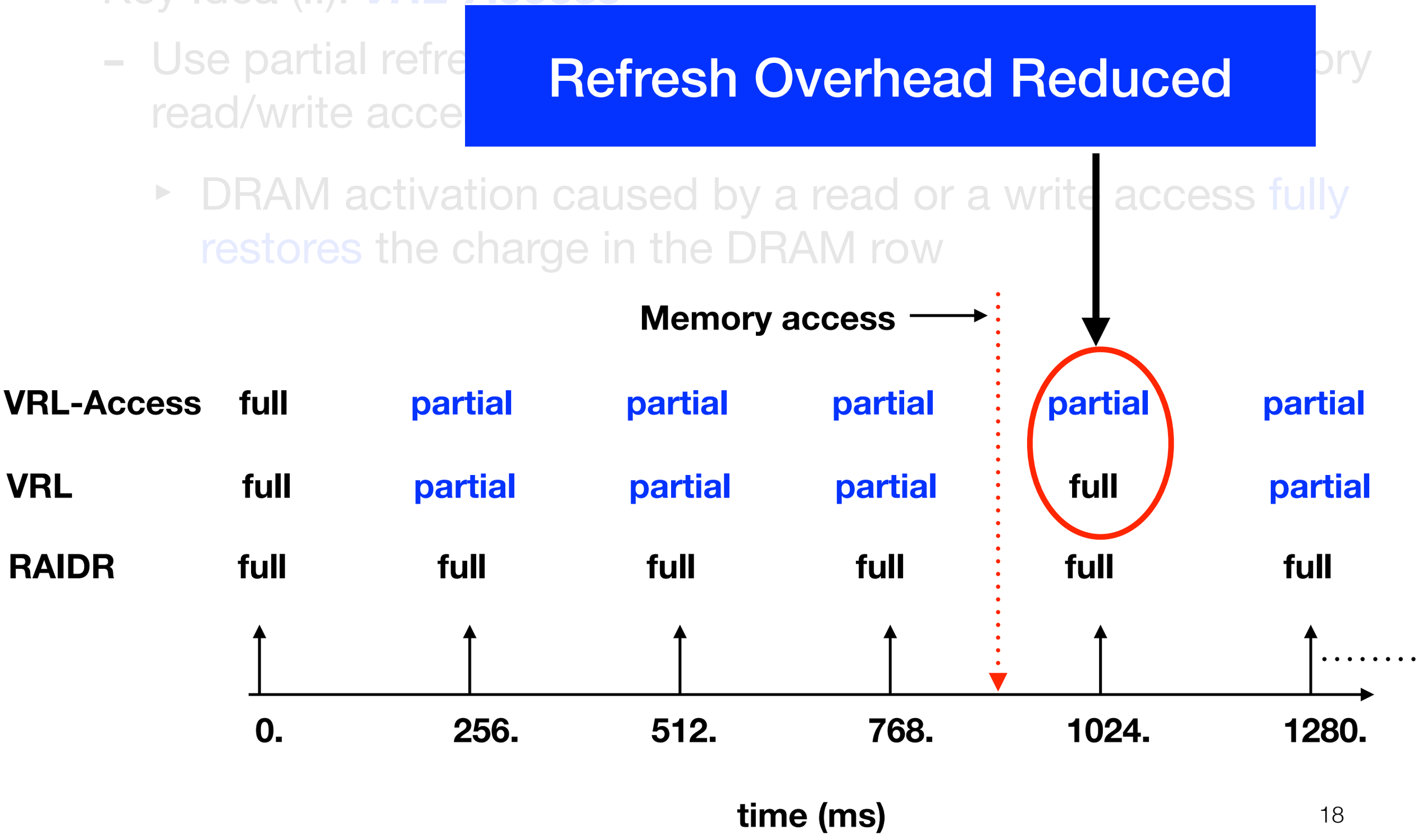
Partial Refresh Scheduling (II)

- Key Idea (II): ***VRL-Access***
  - Use partial refresh in place of full refresh if there is a memory read/write access
    - DRAM activation caused by a read or a write access fully restores the charge in the DRAM row



Partial Refresh Scheduling (II)

- Key Idea (II): ***VRL-Access***
  - Use partial refre… ory read/write acce…
    - DRAM activation caused by a read or a write access fully restores the charge in the DRAM row



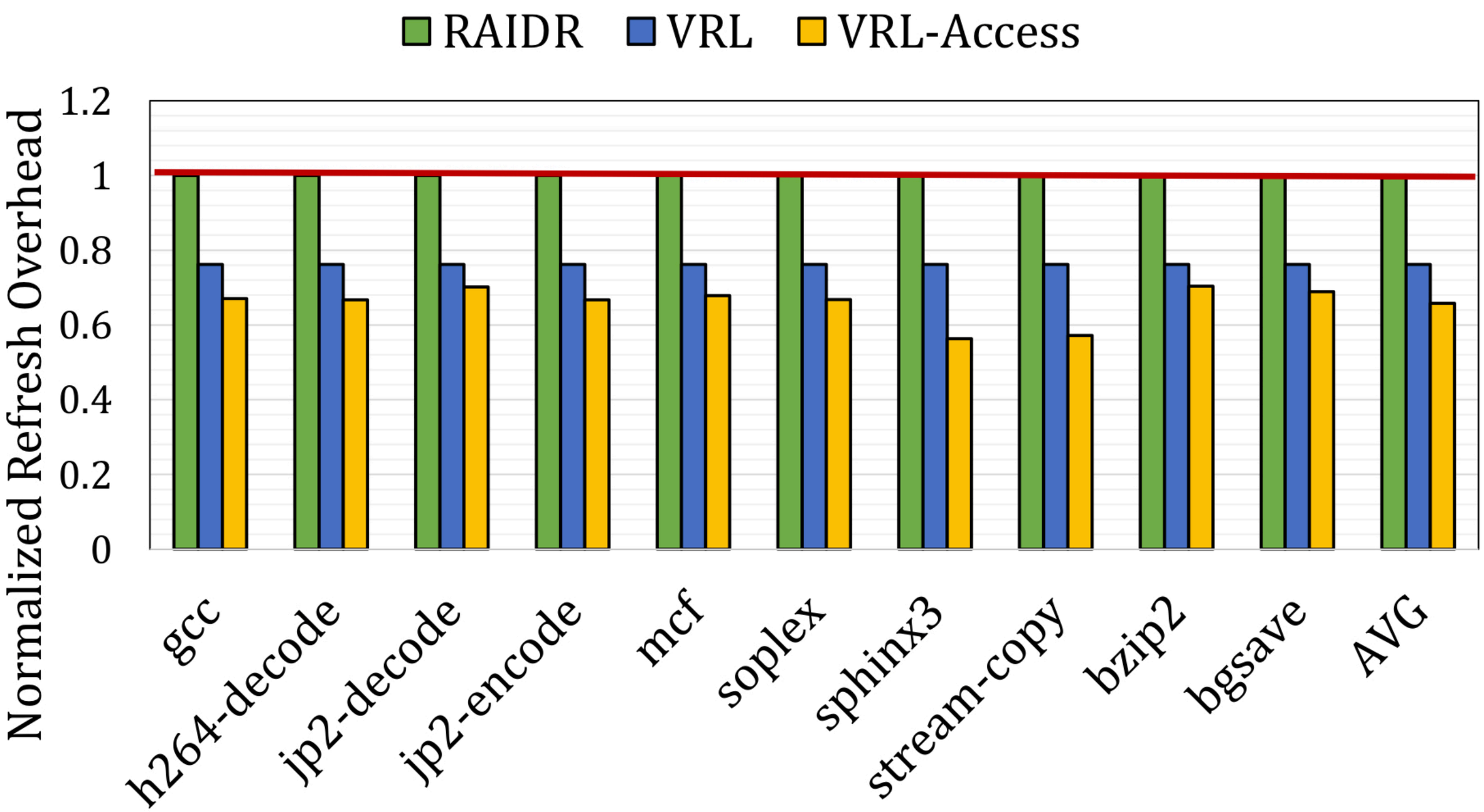
Outline

- Introduction
- DRAM refresh characterization
- Partial and full refresh scheduling
- Evaluation
- Conclusion

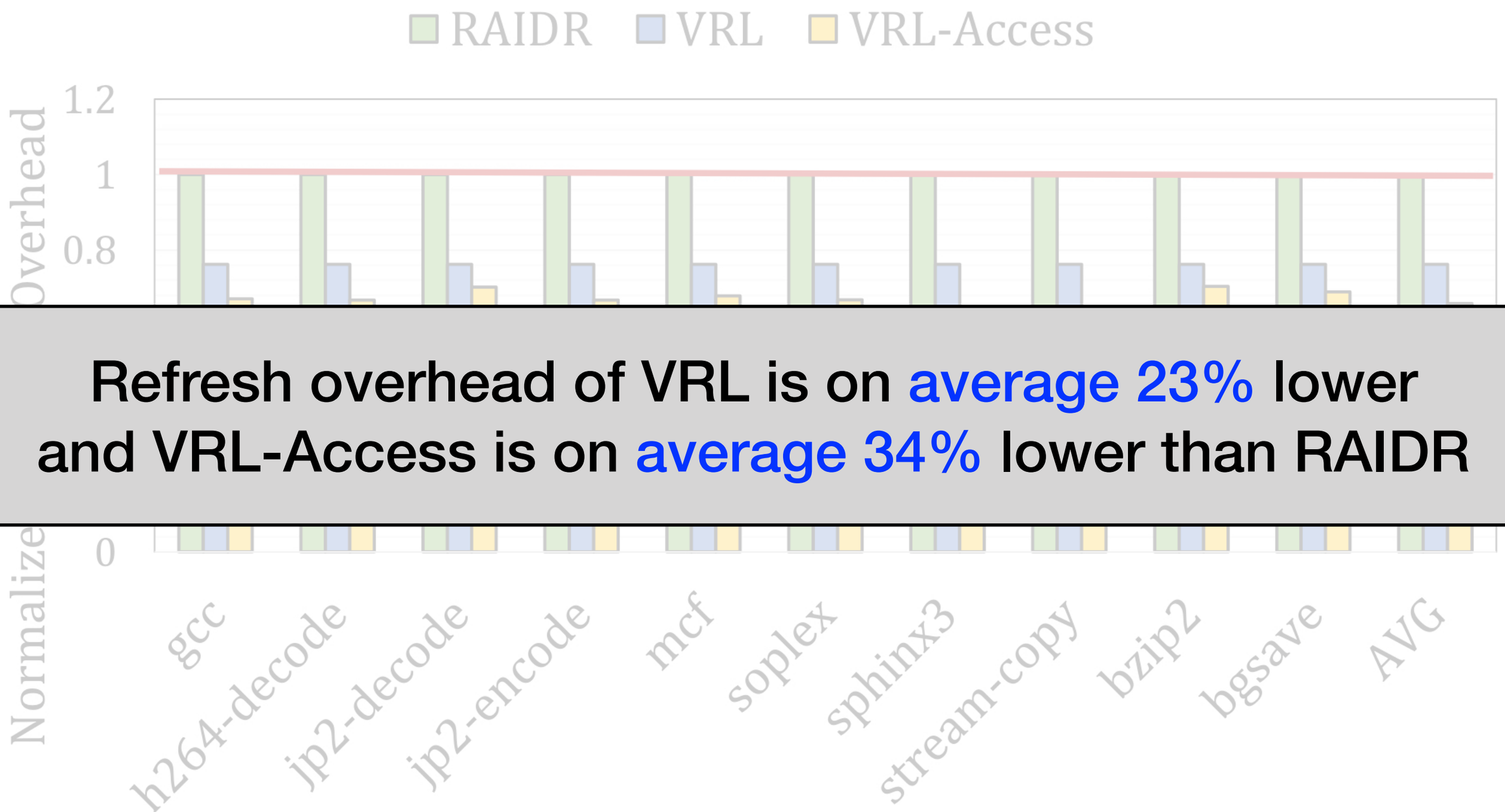
Evaluation Methodology

- PARSEC-3.0 workloads and a server workload *bgsave*
- 90nm technology node for the refresh parameters
- Refresh overhead computed as the fraction of time in every refresh interval that a DRAM is unavailable to service any access request
- Simulate workloads to compute the average refresh overhead for all memory requests

VRL and VRL-Access Reduce Refresh Overhead



VRL and VRL-Access Reduce Refresh Overhead



Refresh overhead of VRL is on average 23% lower and VRL-Access is on average 34% lower than RAIDR

VRL-DRAM Area Overhead

- 90nm technology node from Microwind3

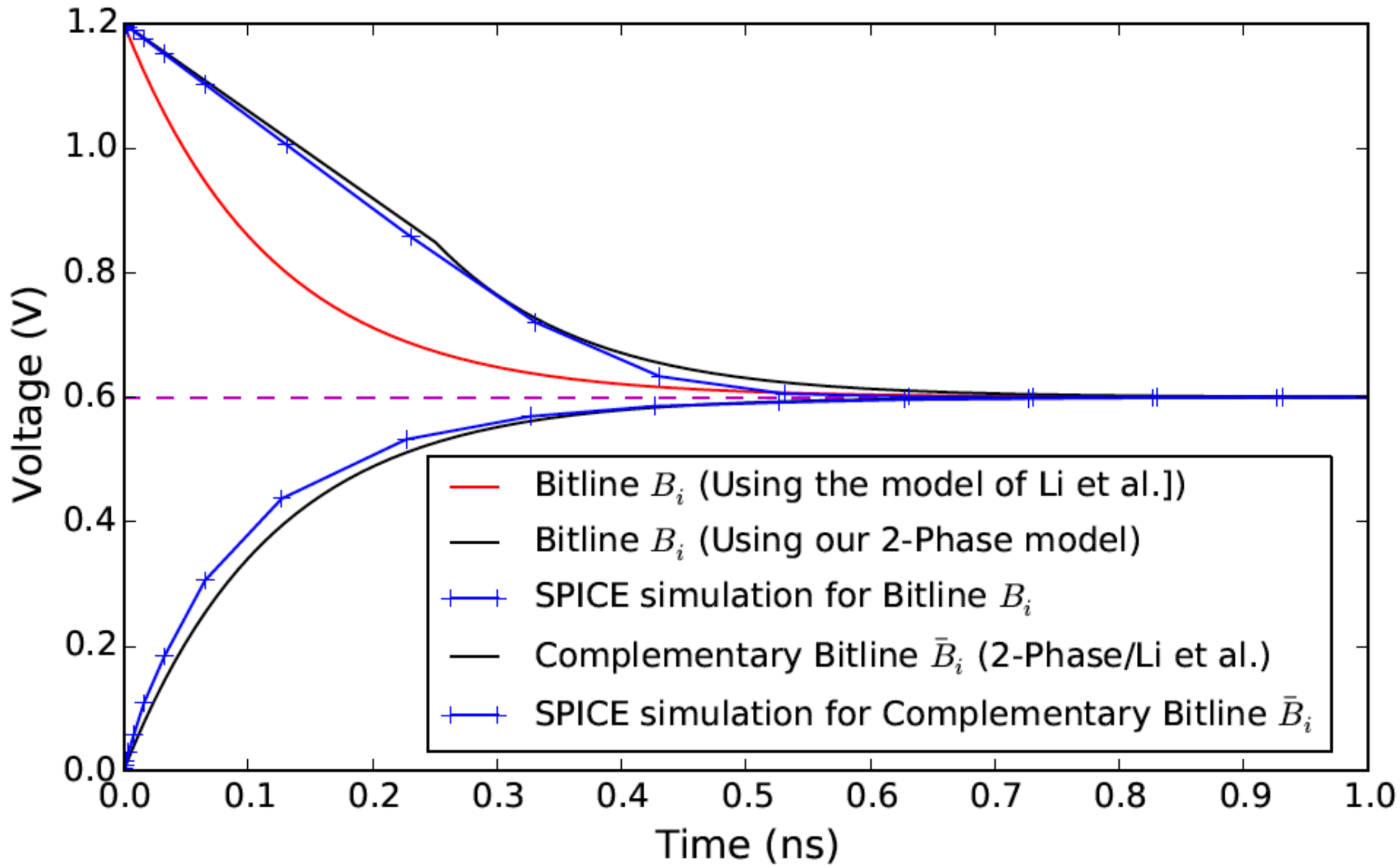
nbits	Logic area (μm^2)	% DRAM bank area (μm^2)
2	105	0.97%
3	152	1.4%
4	200	1.85%

- nbits is the number of bits required to store the number of partial refreshes for a row

- Area overhead $< 2\%$ of DRAM bank area

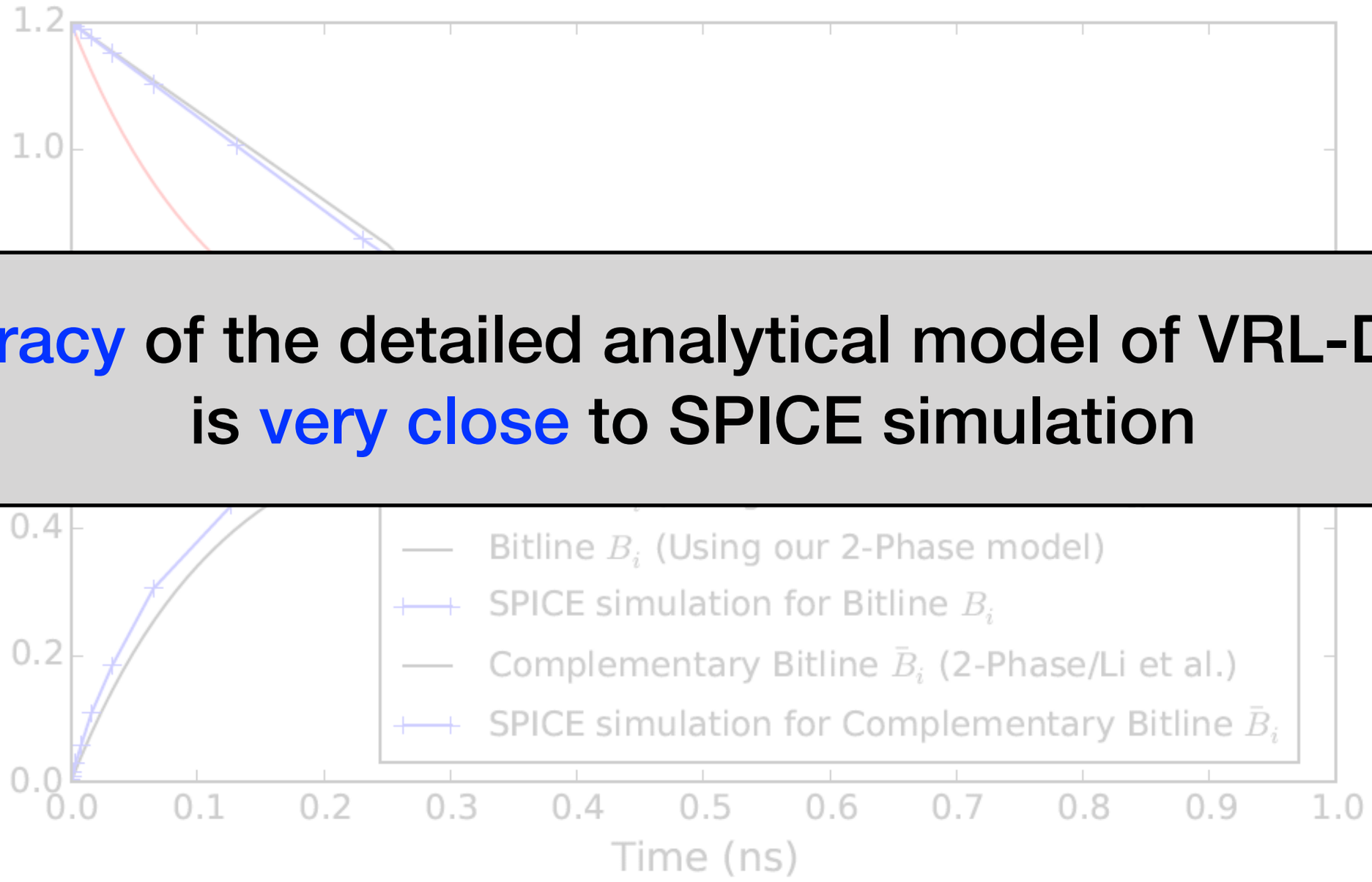
Accuracy of VRL-DRAM tool

- Detailed circuit-level analytical model
  - Tool: *https://github.com/anupkdas-nus/VRL-DRAM*



Accuracy of VRL-DRAM tool

- Detailed circuit-level analytical model
  - Tool: *https://github.com/anupkdas-nus/VRL-DRAM*



Accuracy of the detailed analytical model of VRL-DRAM is very close to SPICE simulation

Outline

- Introduction
- DRAM refresh characterization
- Partial and full refresh scheduling
- Evaluation
- Conclusion

Conclusion

- Observations
  - During refresh, *almost half* of the refresh time is spent in injecting the *last 5% of charge* of a fully charged cell
  - Once *fully* restored, a DRAM cell can sustain multiple *partial* refreshes *without* sacrificing data integrity
- Idea: Variable Refresh Latency DRAM
  - Fully refresh a DRAM cell only when necessary, and otherwise issue partial refresh → reduces refresh latency in common case
- Characterization: new detailed circuit-level analytical model
  - Great potential to lower DRAM refresh timing parameters
- Performance Evaluation
  - Significant refresh overhead reduction (23% for PARSEC-3.0 workloads without errors)

Improving DRAM Performance via Variable Refresh Latency

VRL-DRAM

Anup Das
Hasan Hassan and Onur Mutlu