Flikker: Saving DRAM Refresh Power through Critical Data Partitioning

Karthik Pattabiraman,
with Song Liu,
Thomas Moscibroda,
Benjamin Zorn

Motivation: Smartphones

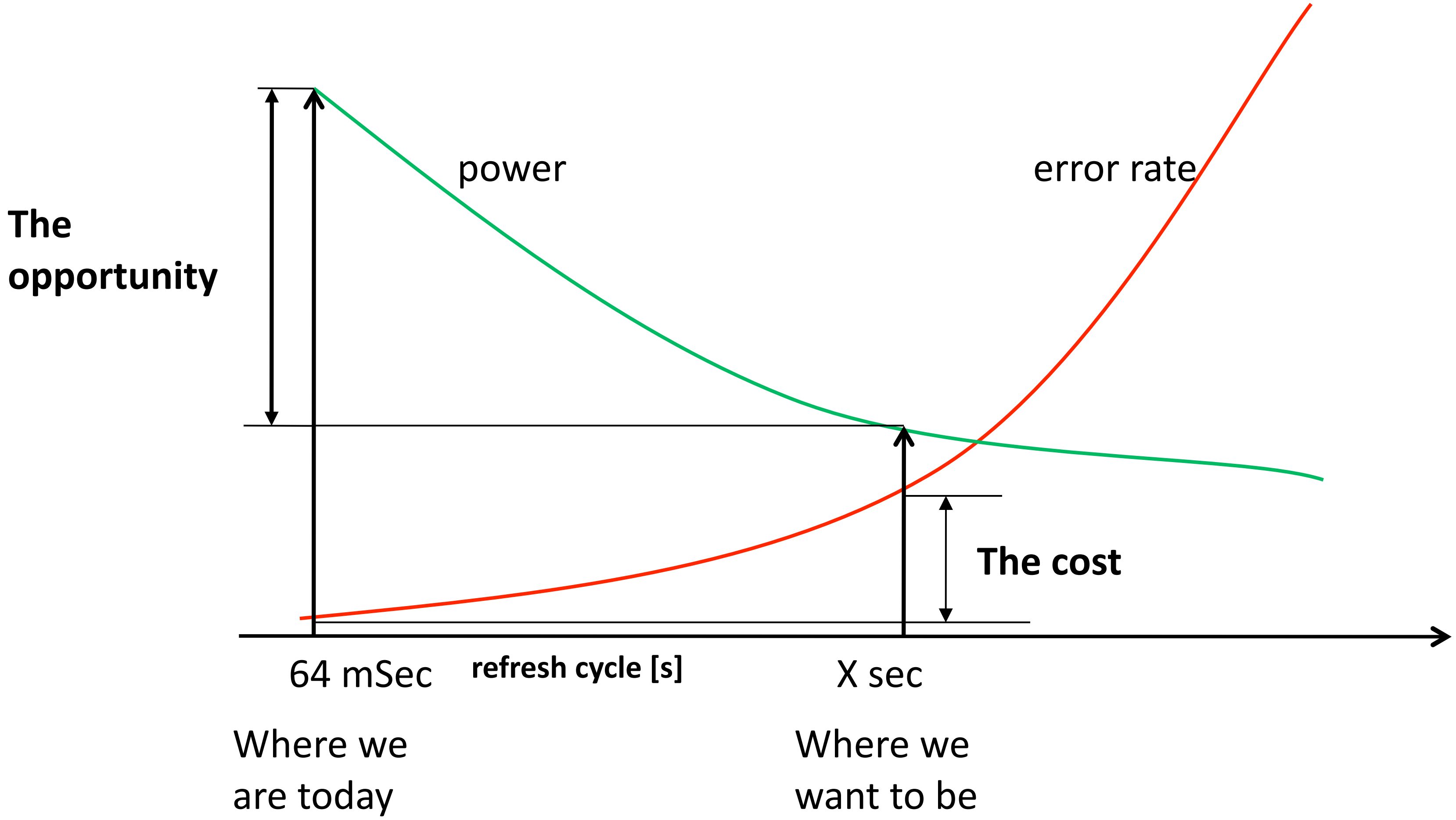
Smartphones becoming ubiquitous

Responsiveness is important

DRAM Memory consumes up to 30% of power

Can drain the battery even when idle

Motivation: DRAM Refresh



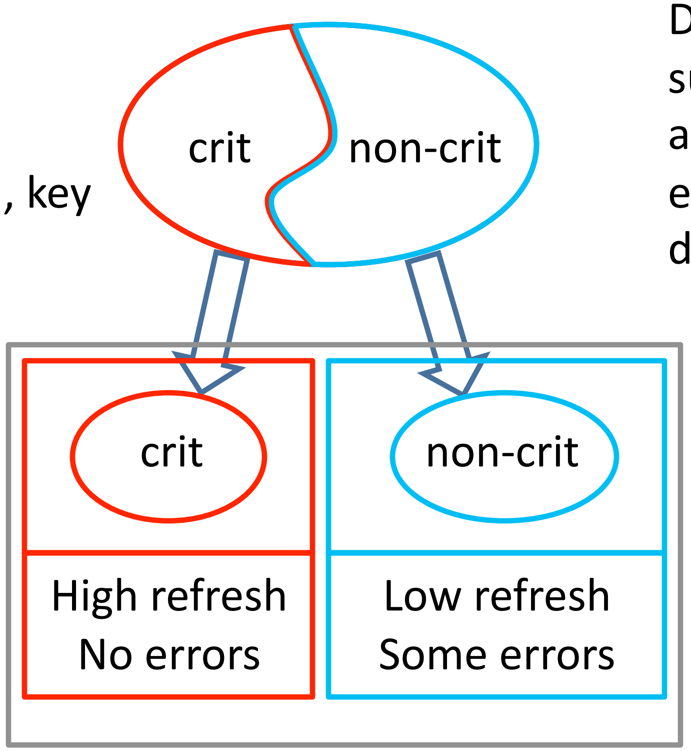
If software is able to tolerate errors, we can lower DRAM refresh rates to achieve considerable power savings

Flikker: Approach

- **Critical / non-critical data partitioning**

Important for application correctness
e.g., meta-data, key data structures

Does not substantially impact app correctness
e.g., multimedia data, soft state

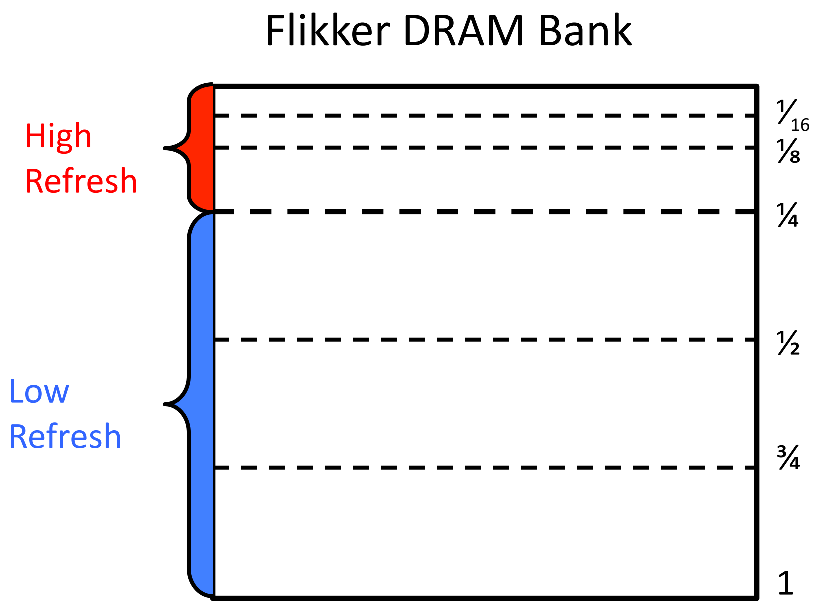


Flikker DRAM

Mobile applications have substantial amounts of non-critical data that can be easily identified by application developers

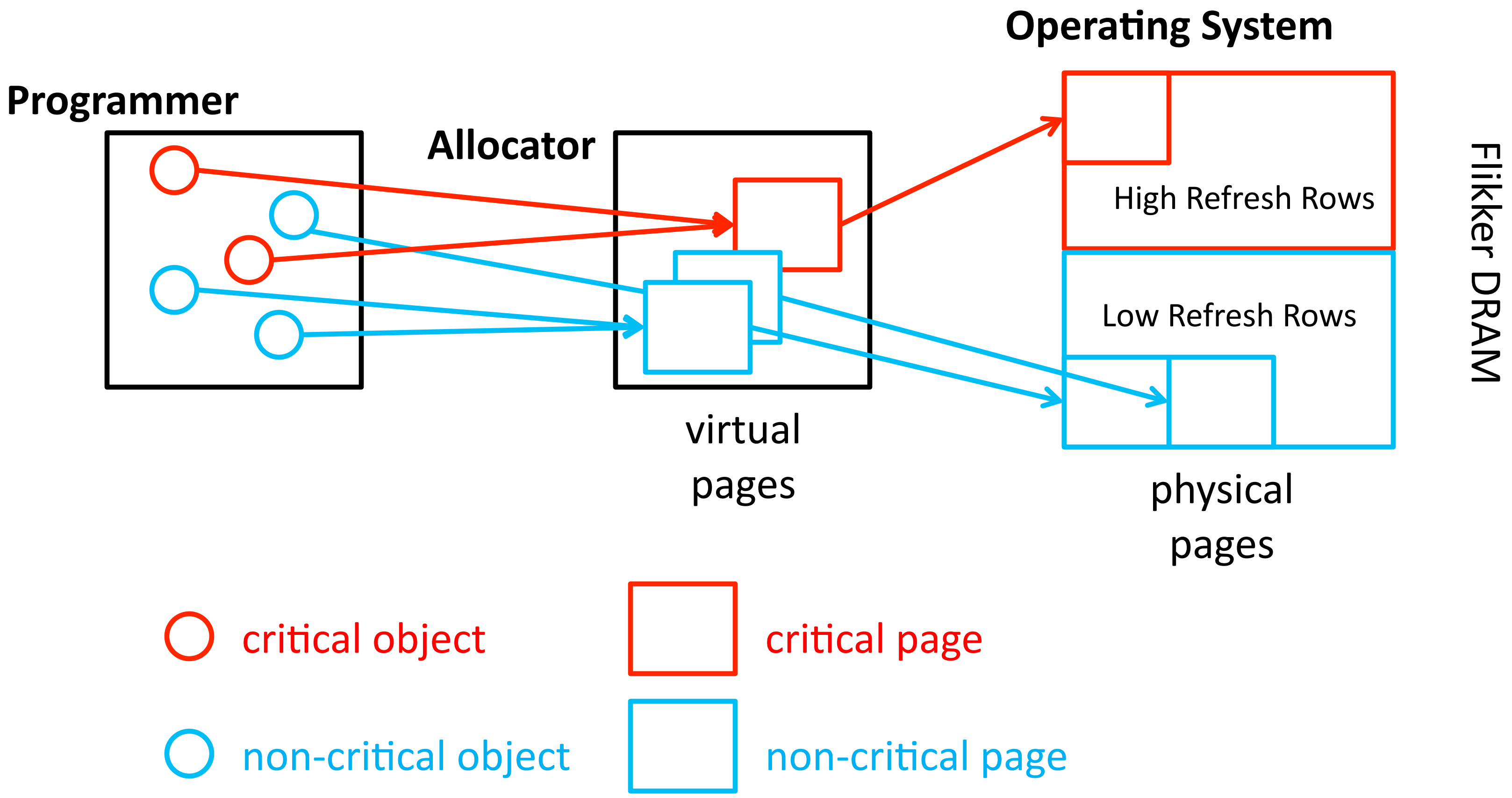
Flikker: Hardware Implementation

- Divide memory bank into high refresh part and low refresh parts
- Size of high-refresh portion can be configured at runtime
- Small modification of the Partial Array Self-Refresh (PASR) mode



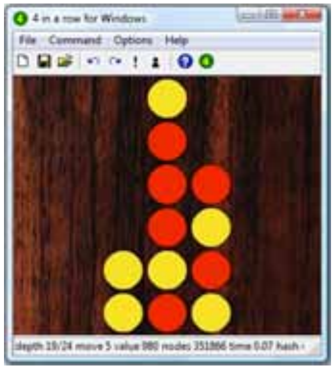
Flikker: Software Implementation

Minor changes to the memory allocator and the Operating System (OS)

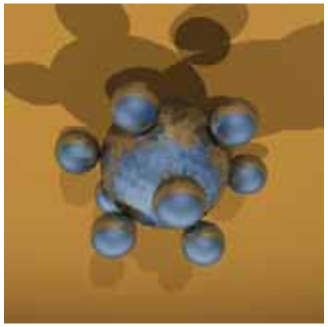


Evaluation: Mobile Applications

- mpeg2 (video decoding)
- c4 (connect 4, four-in-a-row)
- rayshade (ray-traced images)
- vpr (Stochastic optimization)
- parser (Natural-language processing)



```
depth 19/24 move 5 value 980 nodes 351866 time 0.07 hash
```



Application	No. of lines	Input	Metric
mpeg2	10.0k	mei16v2.m2v	output SNR
c4	6.1k	N/A	saved moves
rayshade	24.2k	balls.ray	output SNR
vpr	24.6k	ref/test	output file
parser	11.5k	ref/test	output file

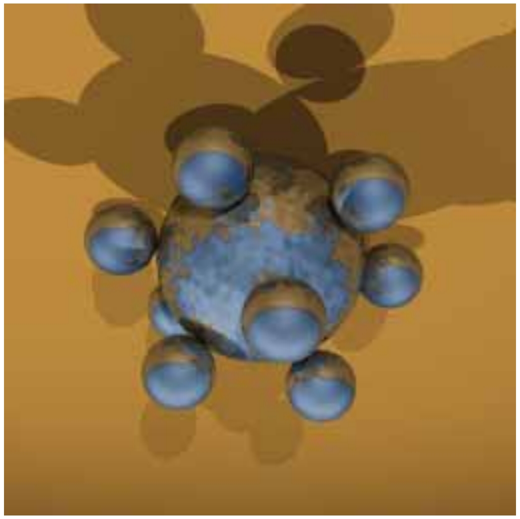
Evaluation: Summary

- **Performance overhead**: < 1 % **(real system)**

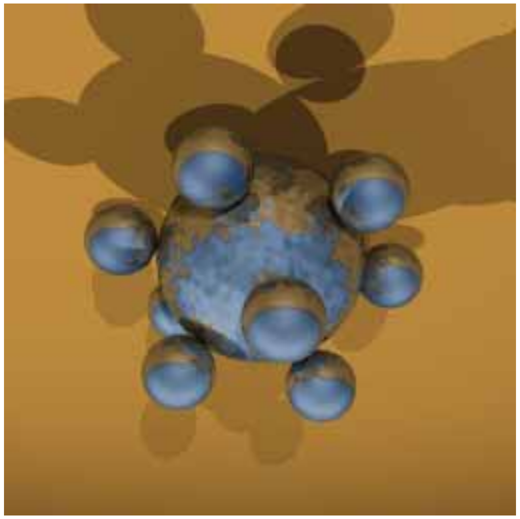
- **Power savings (evaluated using simulation)**
  - 30 to 35% of standby power reduction
  - 20-25% of overall DRAM power reduction

- **Reliability (evaluated using fault-injection)**
  - **No effect for c4, vpr, and parser applications**
    - But crashes and incorrect outputs occur without Flikker
  - **Some degradation of SNR for mpeg2 and ray-shade**
    - SNR reduced from over 100 dB to 78.9 db for Rayshade
    - SNR reduced marginally for the mpeg2 decoder

Rayshade: Degraded SNR

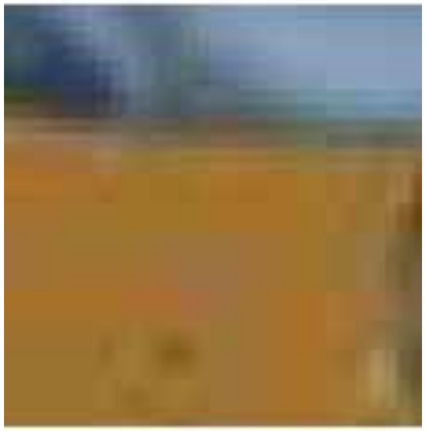


Original



78.9dB

2 X Zoom



Flikker: Summary

- **First software technique to intentionally lower hardware memory reliability for energy savings**
  - Minimal changes to hardware – based on PASR mode
  - Minor changes to applications to identify critical data

- Reduced the overall DRAM memory power by **20-25%** with negligible loss of performance and reliability across five mobile applications

- **Future work:** Extension to data center applns.

The “Good Enough” Revolution

Source: WIRED Magazine (Sep 2009) – Robert Kapps
http://www.wired.com/gadgets/miscellaneous/magazine/17-09/ff_goodenough

People prefer “cheap and good-enough” over “costly and near-perfect”

Can we design dependable systems with this principle ?

More Information …

- See our upcoming paper at **ASPLOS'2011**

http://synergy.ece.ubc.ca/karthik/

Flikker: Saving DRAM Refresh Power through Critical Data Partitioning,

Song Liu, Northwestern University

Karthik Pattabiraman, Univ of British Columbia

Thomas Moscibroda, Microsoft Research

Benjamin Zorn, Microsoft Research

Flicker: Related Work

- Better-Than-Worst-Case (BTWC) design
  - Razor [Austin'04]: Save processor power

- Reduce refresh rate and handle faults in leaky rows
  - Do not use faulty rows [ESKIMO Micro'09]
  - Refresh different rows at different rate [Kim TVLSI'03] [Venkatesan - HPCA'06]
  - Only refresh necessary rows [Ghosh MICRO'07]
  - Use ECC [Katayama DFT'99]

Motivation: Hardware Memory Errors

- Memory elements are susceptible to soft-errors (cosmic ray strikes, alpha particles etc.)
- Variation in retention times among DRAM cells
  - Anywhere from a few milli-seconds to a few seconds

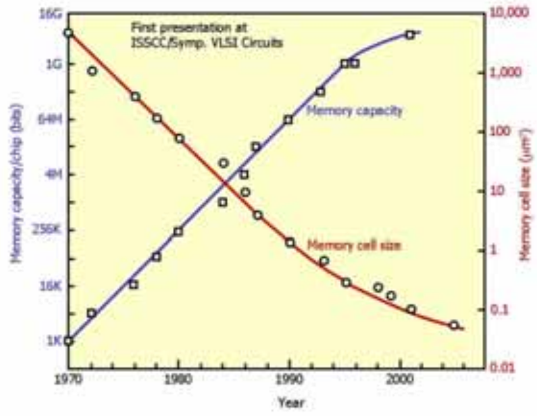


Figure 1

Figure from [Itoh'08]

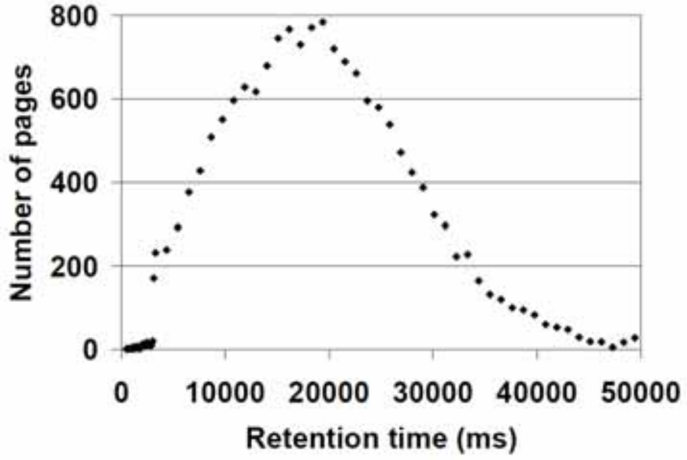
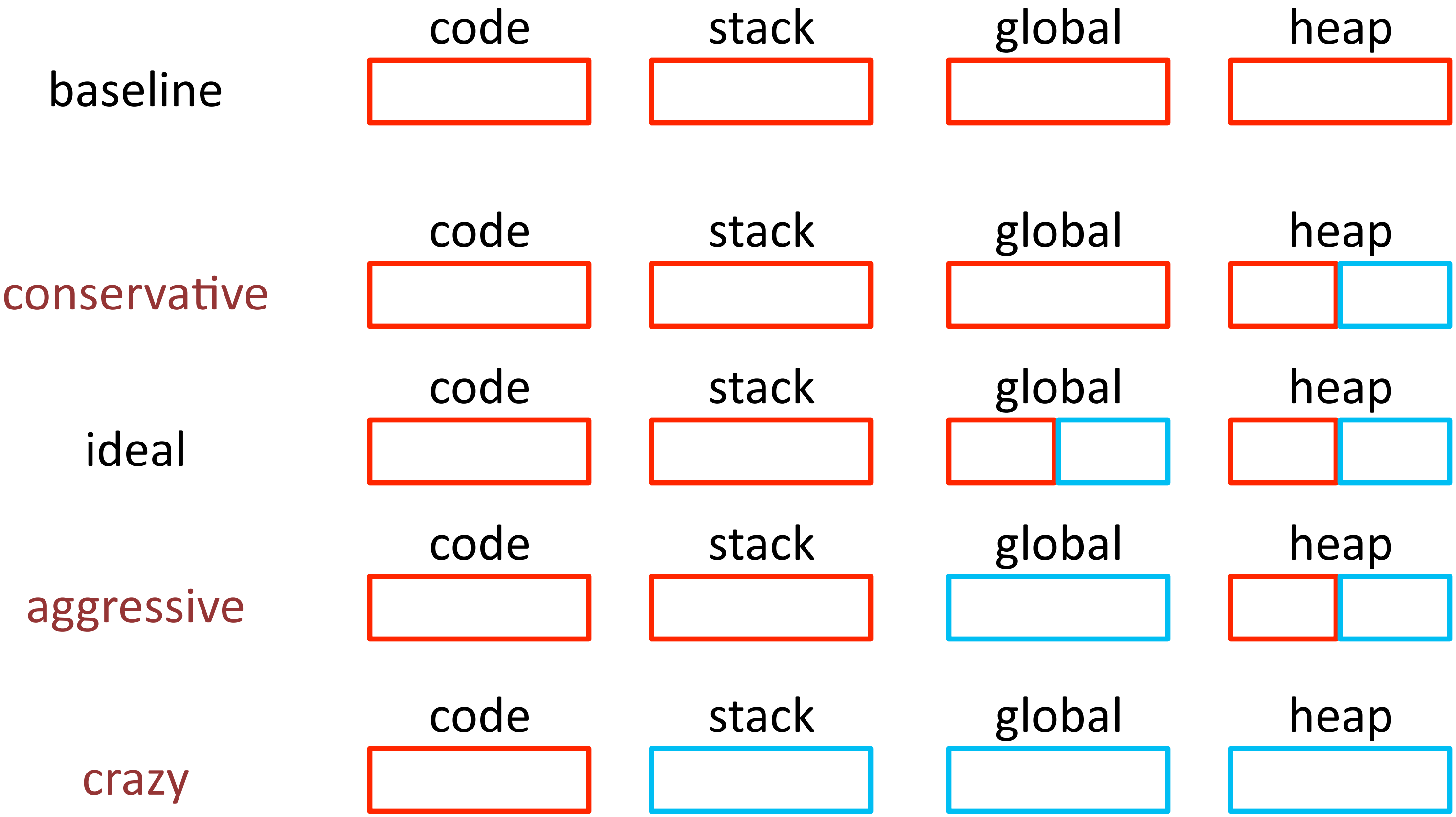


Figure from [Venkatesan'06]

Flikker: Configurations

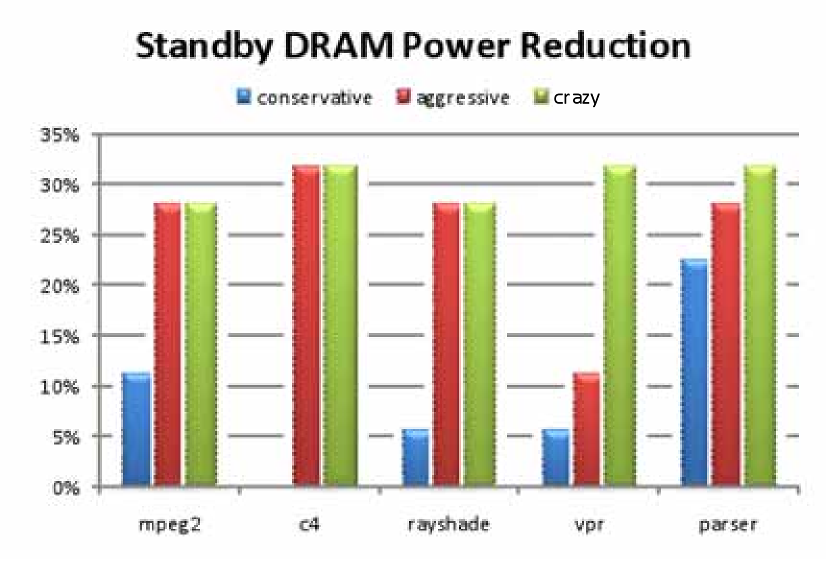


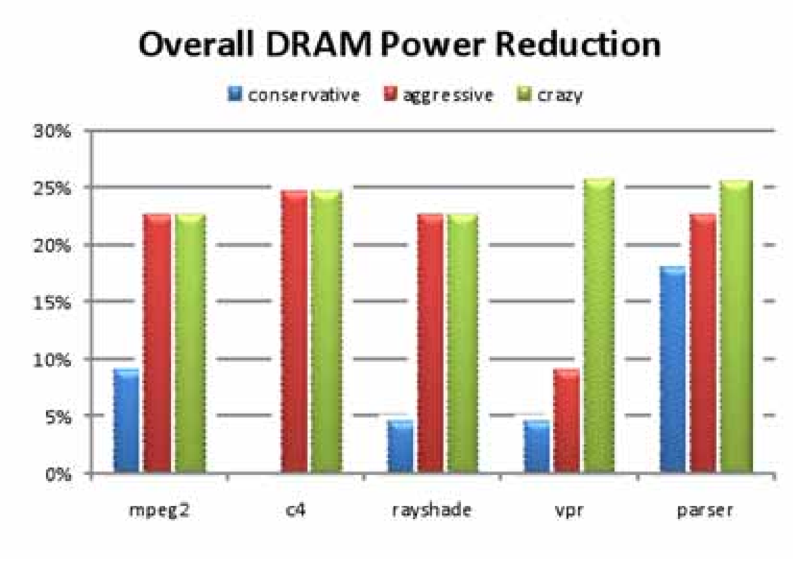
	code	stack	global	heap
baseline	🟥	🟥	🟥	🟥
conservative	🟥	🟥	🟥	🟥🟦
ideal	🟥	🟥	🟥🟦	🟥🟦
aggressive	🟥	🟥	🟦	🟥🟦
crazy	🟥	🟦	🟦	🟦

Flikker: Power Reduction

- Standby power: analytical model

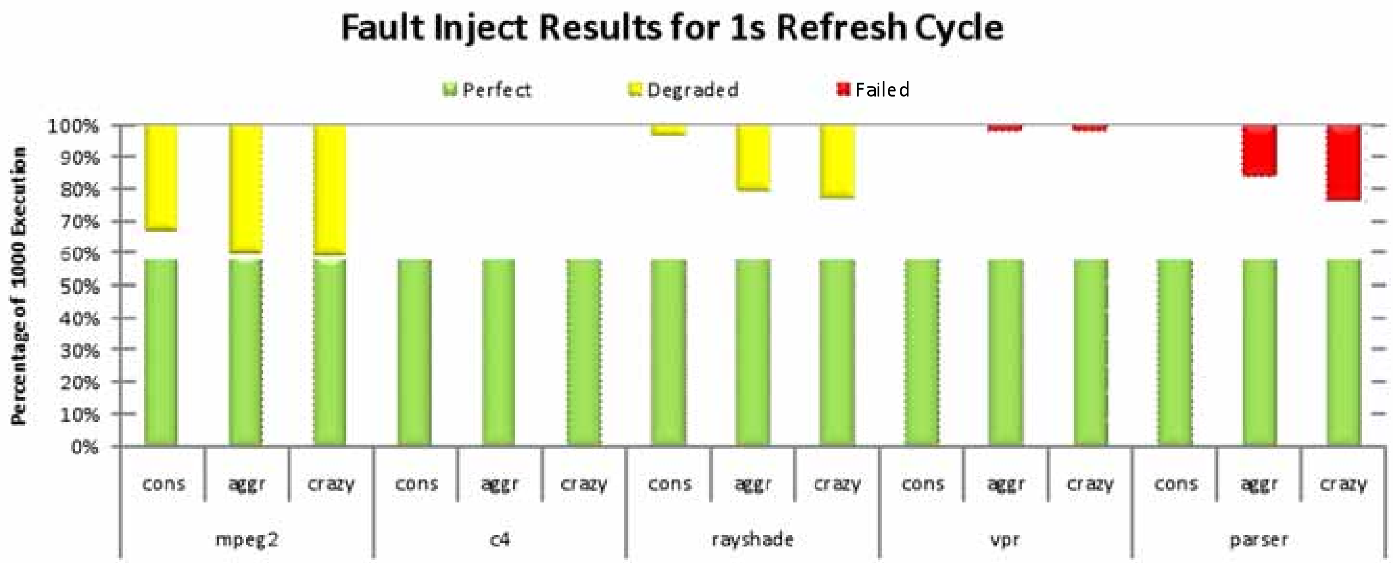
□ Overall power: analytical model, simulation, usage profile (5% active, 95% standby) [Karlson et.al, 2009]





Fault-injection Result: 1s refresh

- Output stats (1000 executions): perfect, degraded, failed
- c4: always perfect outputs
- mpeg2, rayshade: some degraded in aggressive and crazy
- vpr, parser: some failed in aggressive and crazy



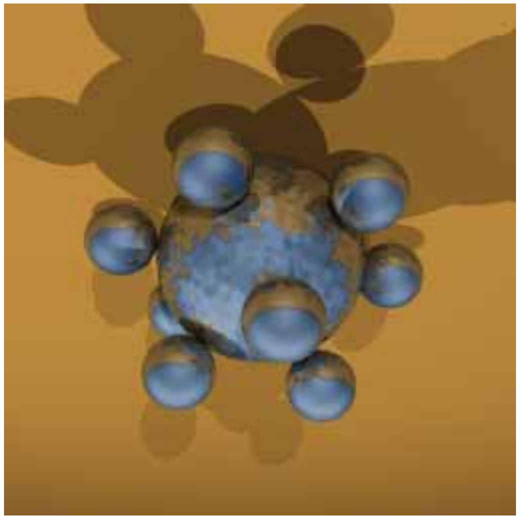
Fault-injection Result: SNR

- Signal-to-Noise-Ratio (SNR): the ratio of signal energy and noise energy
- SNR is logarithm scale: 3dB means double in energy
- mpeg2 encoder -> decoder: 35 dB
- Flicker yields very high SNR

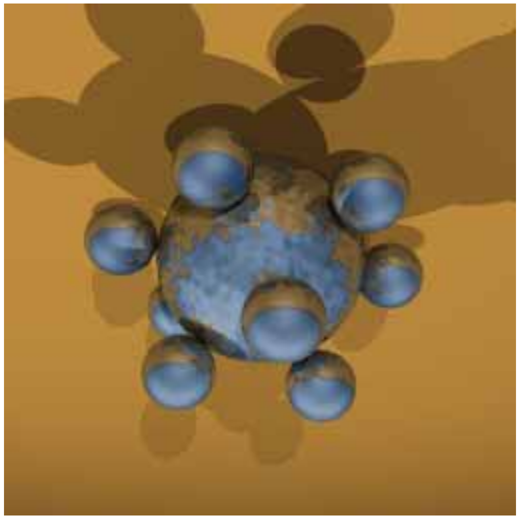
Configuration	mpeg2	rayshade
conservative	95.48	101.1
aggressive	88.34	72.84
crazy	88.04	73.63

Average SNR of degraded output of mpeg2 and rayshade [dB].

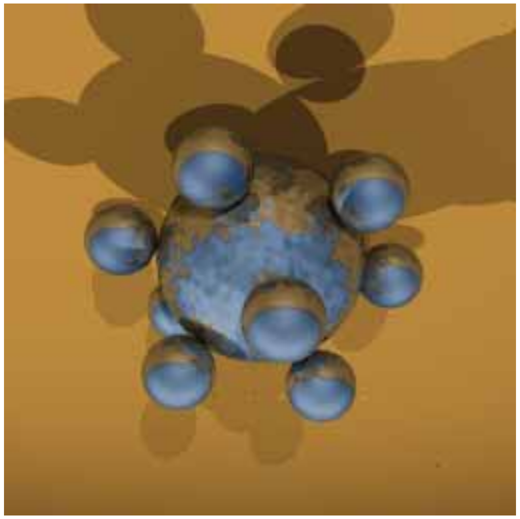
Rayshade Output with Different SNR



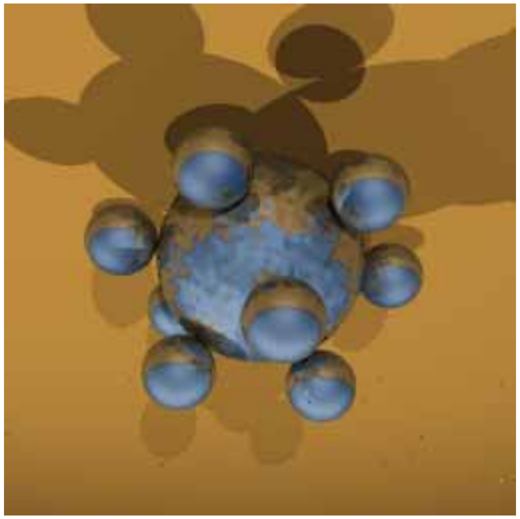
Original



78.9dB



52.0dB



41.3dB