How do Big Systems Fail? (and what can we do about it?)

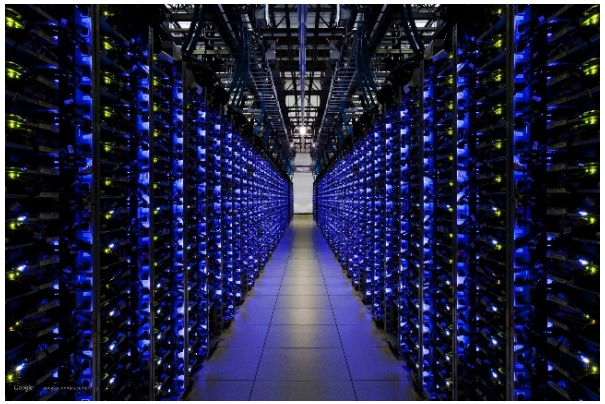
Ioan Stefanovici
Microsoft Research

Imperial College

Feb. 14, 2017

Big Systems?

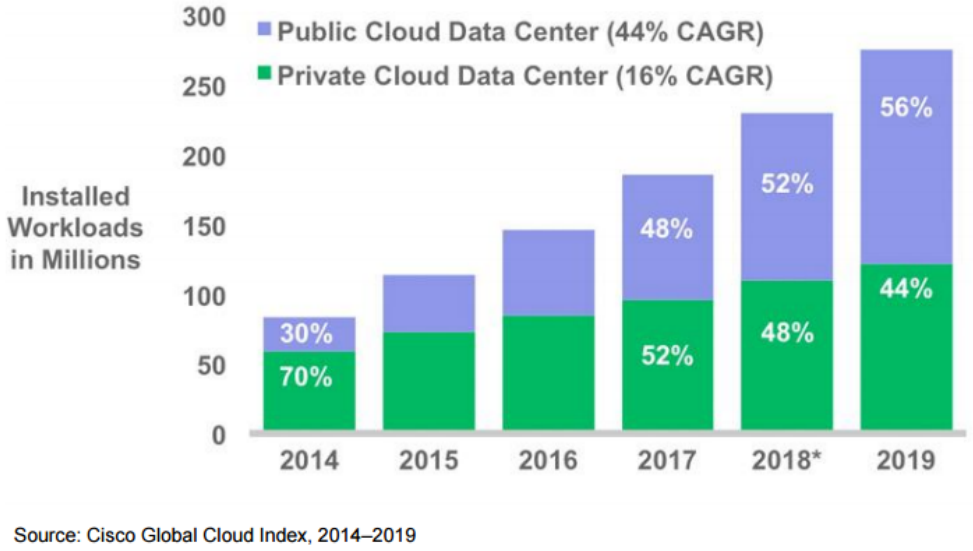
- Data centers



- Supercomputers


Who cares?

- Supercomputers: the government
- Data center usage is soaring

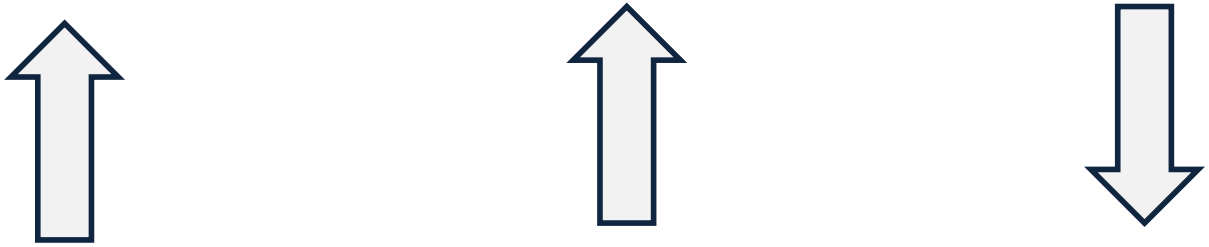


Source: Cisco Global Cloud Index, 2014–2019

- World's financial data
- E-commerce sales information
- Medical data
- Scientific data
- Personal
  - E-mail
  - Text & video messaging
- Music, videos, photos
- *App* data
- …

- > 90% of new data generated today is stored digitally
- ~ 44 trillion gigabytes of data by 2020
- Cornerstone of our modern life

Why study reliability?



Demand on systems | Component count | Reliability

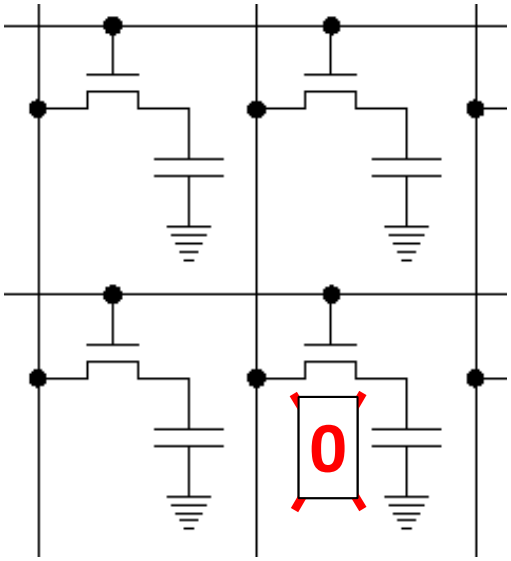
- Simple example:
  - Your laptop/desktop crashes 2 times a year (generous…)
  - What happens when you put 80,000 of them together?
    - 438/day → 18/hour → failure every 3.2 mins
- Large-scale system reliability is not well understood
  - Error models are based on simplistic/theoretical assumptions
  - Shortage of data from *real, productions* systems

Why DRAM errors?

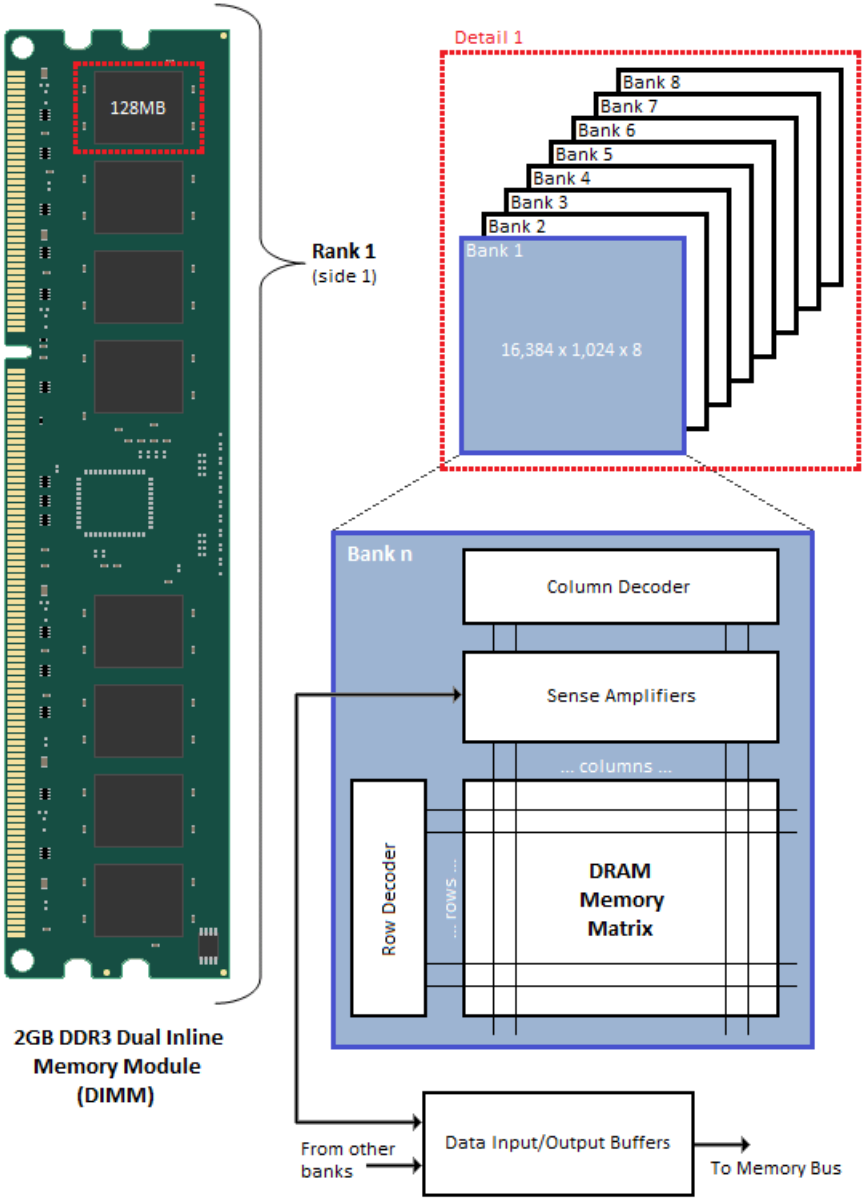
- Why DRAM?
  - One of the most frequently replaced components
  - Getting worse in the future?

- DRAM errors?
  - A bit is read differently

    from how it was written



Memory Addressing



Rank

↓

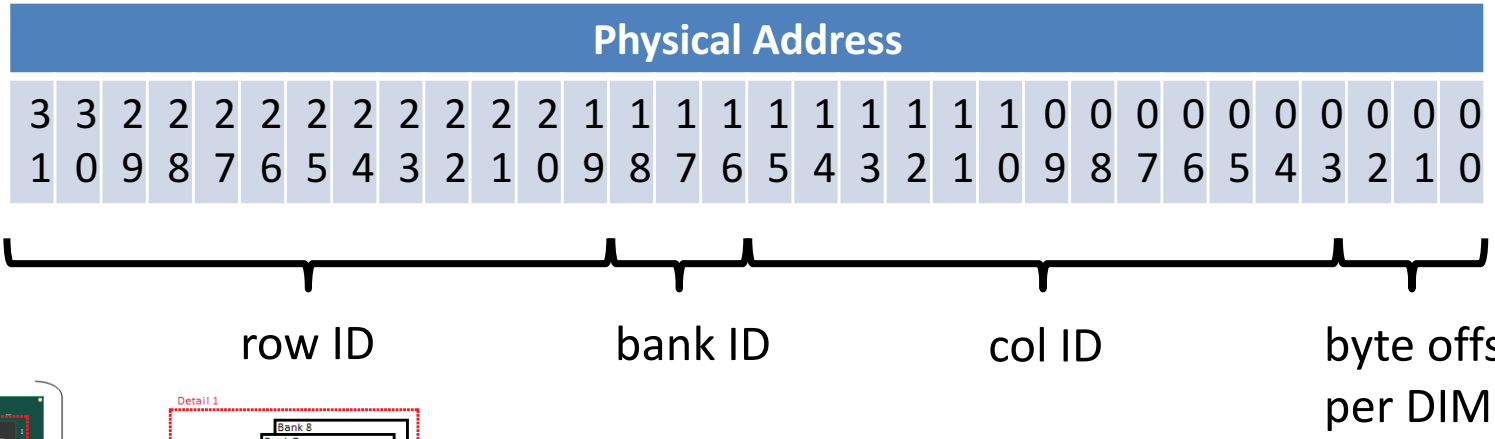
Bank

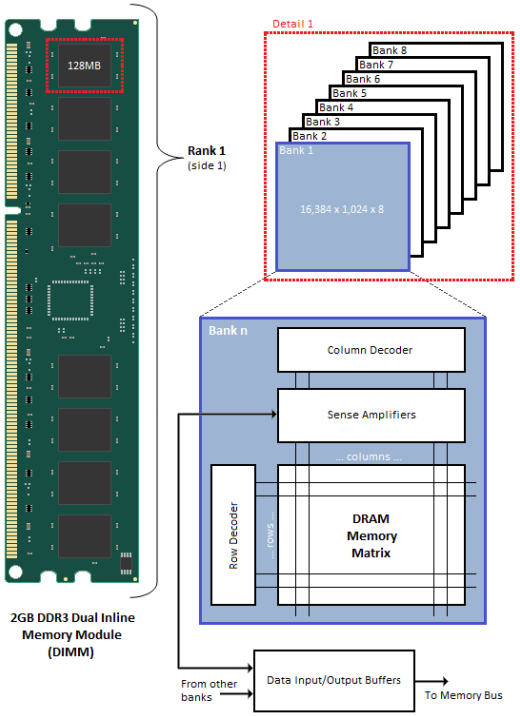
↓

(Row, Column)

(Img source: http://www.anandtech.com/show/3851/everything-you-always-wanted-to-know-about-sdram-memory-but-were-afraid-to-ask)

Memory Addressing Example



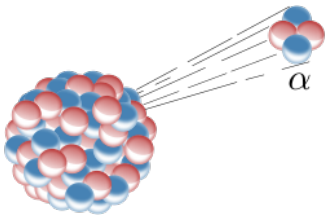


- Physical address "slicing" into bank, row, column done based on:
  - Memory capacity in the system
  - Increasing parallelism, reduce contention
  - Other architectural considerations

(for more, see "Memory Systems: Cache, DRAM, Disk" by Bruce Jacob, David D. Wang, and Spencer Ng.)

Memory Errors

- 1978: Intel's 16-Kbit DRAM devices
  - Abnormally high error rates
- Traced to high-concentration of alpha particles



(2 protons + 2 neutrons)

(https://commons.wikimedia.org/w/index.php?curid=2858666)

- Part of DRAM material manufactured downstream from old uranium mine
- Alpha particles and cosmic rays (after interacting with Earth's atmosphere) cause bit flips ("**soft errors**")
  - …sometimes (and altitude matters)

How do we fix it?

- Parity
  - E.g.: $0 \oplus 0 \oplus 1 \oplus 0 =$ 1
  - $\oplus$ (XOR) counts the # of 1s in the word
  - Add spare bit to hardware to store parity
  - Can only detect an error occurred (can't fix it)
- SEC-DED (Single Error Correct, Double Error Detect)
  - Use Hamming codes
  - Same idea: add extra hardware to store redundancy
- "Chipkill" (IBM)
  - Tolerates failure of an entire DRAM chip
  - Similar to RAID

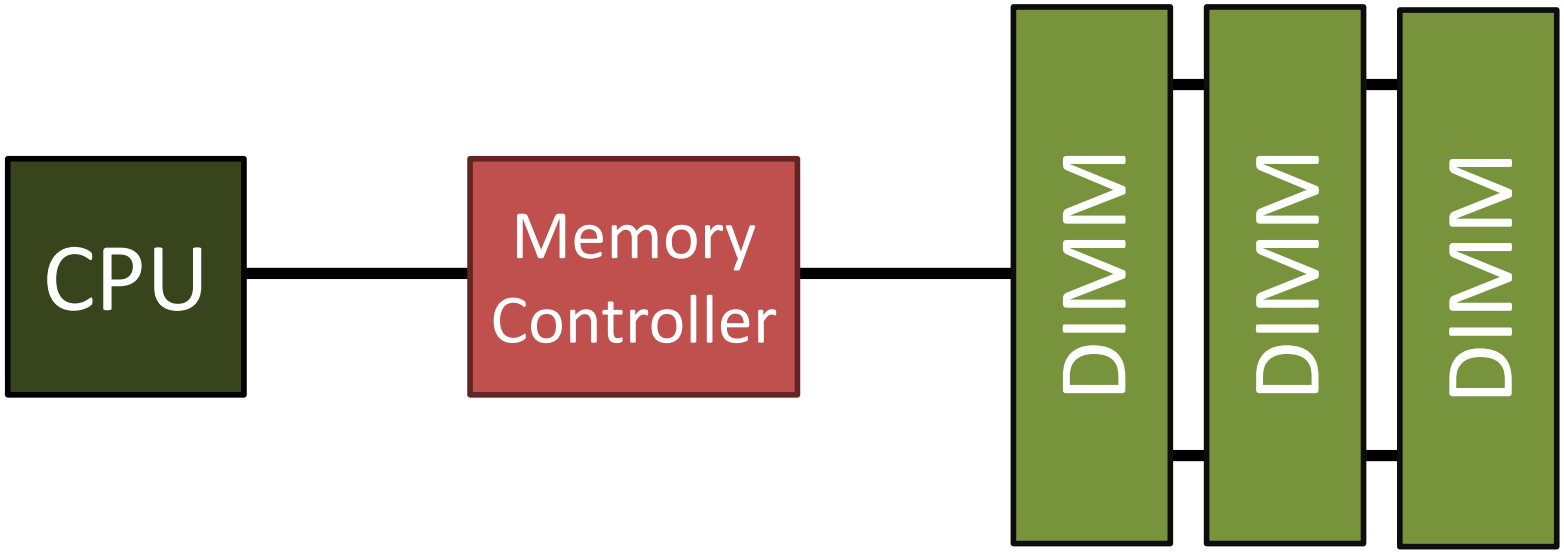
Preventing Data Corruption

- Memory ECC still limited
  - Don’t want errors to accumulate!
  - E.g.: 2 bits enough for corruption with SEC-DED
- Machines today have **lots** of memory
  - Low latency to data
  - Some data may be infrequently accessed
- Solution: memory scrubbers
  - Background HW process: reads + re-computes ECC info
  - If error occurred, fix it + put correct data back in DRAM
- Great! We’re done with soft errors!
  - (except for “**hard errors**” –HW problems-)

State of the World ~ 10 years ago

- Memory errors were a **huge** problem
  - In data centers and supercomputers
- MANY open questions:
  - What does the error process look like? (Poisson?)
  - What is the frequency of hard vs. soft errors?
  - What do errors look like on-chip?
  - Can we predict errors?
  - What is the impact on the OS?
  - How effective are hardware and software level error protection mechanisms?
    - Can we do better?

The data in our study



- Error events detected upon [read] access and corrected by the memory controller
- Data contains error location (node and address), error type (single/multi-bit), timestamp information.

The systems in our study

System	DRAM Technology	Protection Mechanisms	Time (days)	DRAM (TB)
LLNL BG/L	DDR	Multi-bit Correct, Bit Sparing	214	49
ANL BG/P	DDR2	Multi-bit Correct, Chipkill, Bit Sparing	583	80
SciNet GPC	DDR3	SEC-DED	211	62
Google	DDR[1-2], FBDIMM	Multi-bit Correct	155	220

- Wide range of workloads, DRAM technologies, protection mechanisms.
- Memory controller physical address mappings
- In total more than 300 TB-years of data!

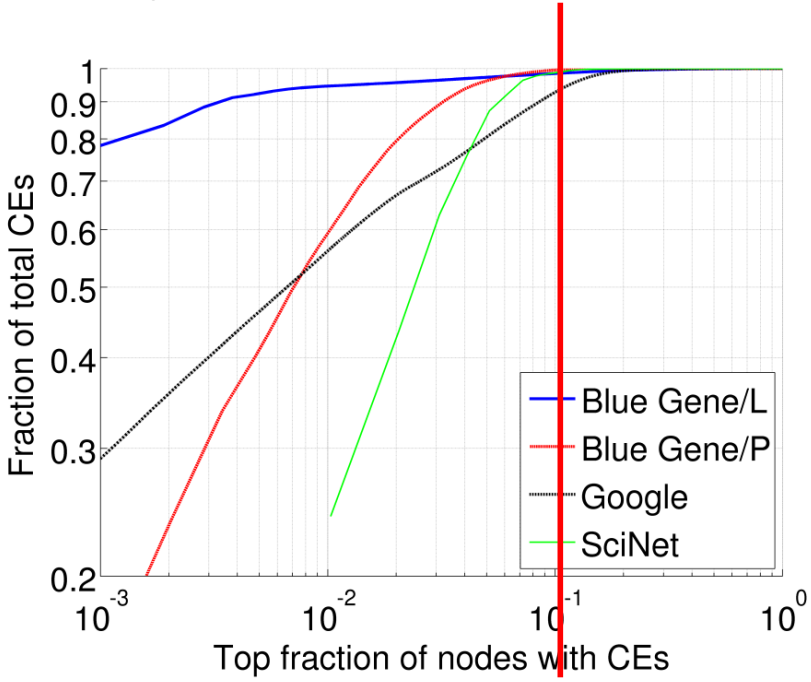
How common are DRAM errors?

System	Total # of Errors in System	Nodes With Errors	Average # Errors per Node / Year	Median # Errors per Node / Year
LLNL BG/L	227×10^6	1,724 (**5.32%**)	3,879	19
ANL BG/P	1.96×10^9	1,455 (**3.55%**)	844,922	14
SciNet GPC	49.3×10^6	97 (**2.51%**)	263,268	464
Google	27.27×10^9	20,000 (**N/A %**)	880,179	303

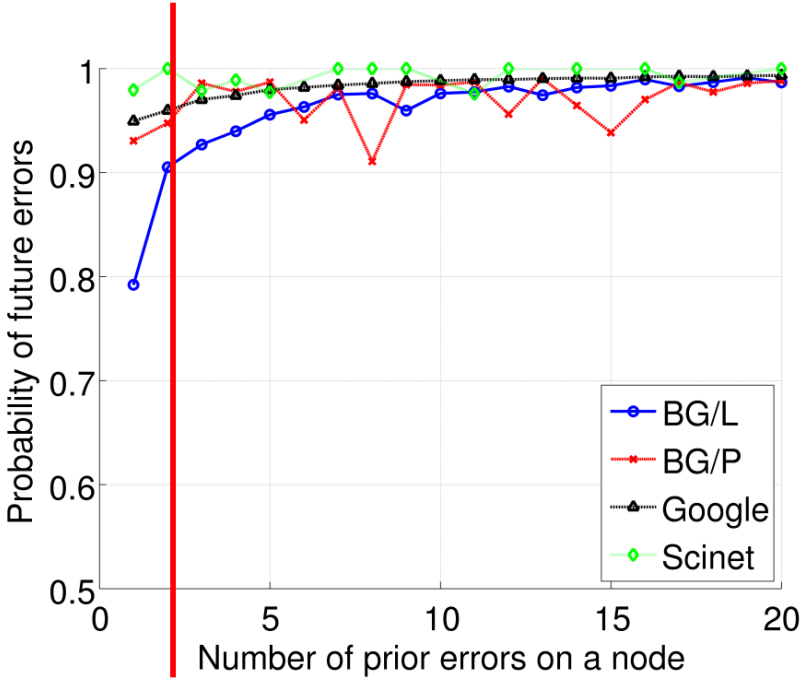
- Errors happen at a significant rate
- Highly variable number of errors per node

How are errors distributed in the systems?

- Only 2-20% of nodes with errors experience a single error
- Top 5% of nodes with errors experience > 1 million errors



Top 10% of nodes with CEs make up ~90% of all errors

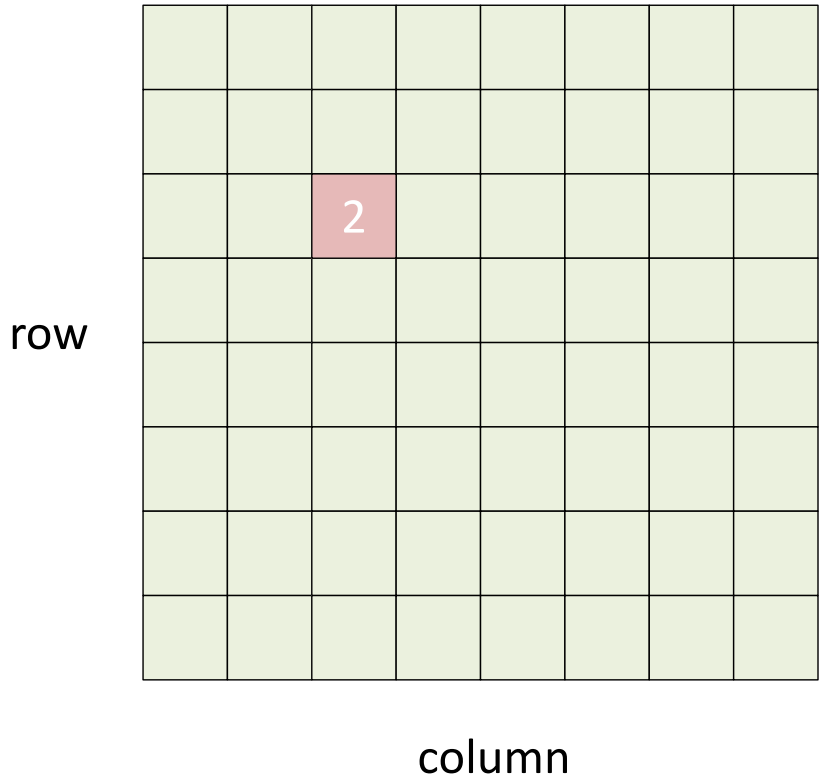


After 2 errors, probability of future errors > 90%

- Distribution of errors is highly skewed
  - Very different from a Poisson distribution
- Could hard errors be the dominant failure mode?

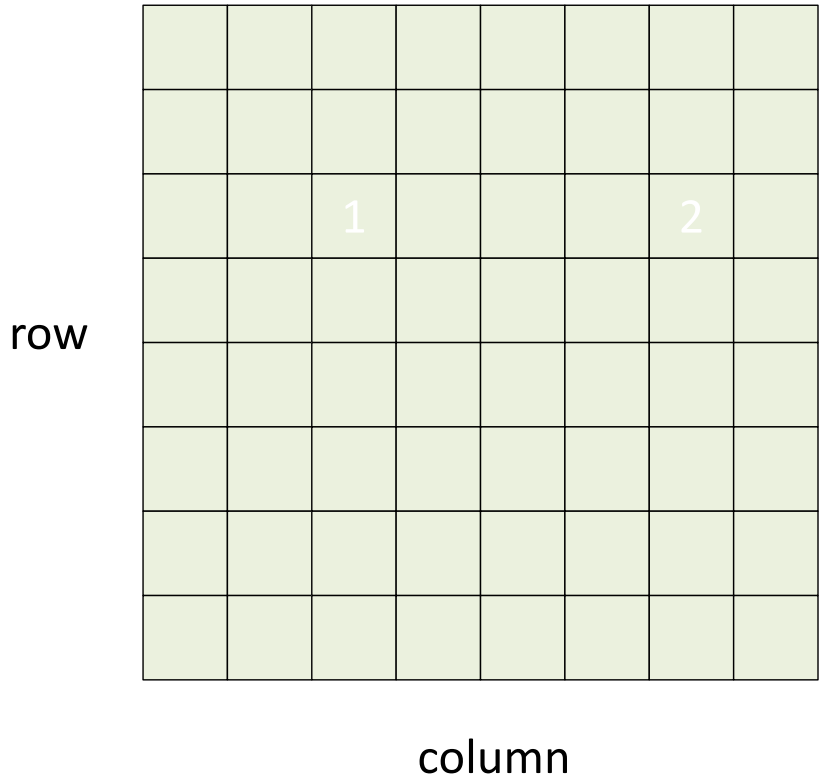
What do errors look like on-chip?

Error Mode
Repeat address
Repeat row
Repeat column
Whole chip
Single Event



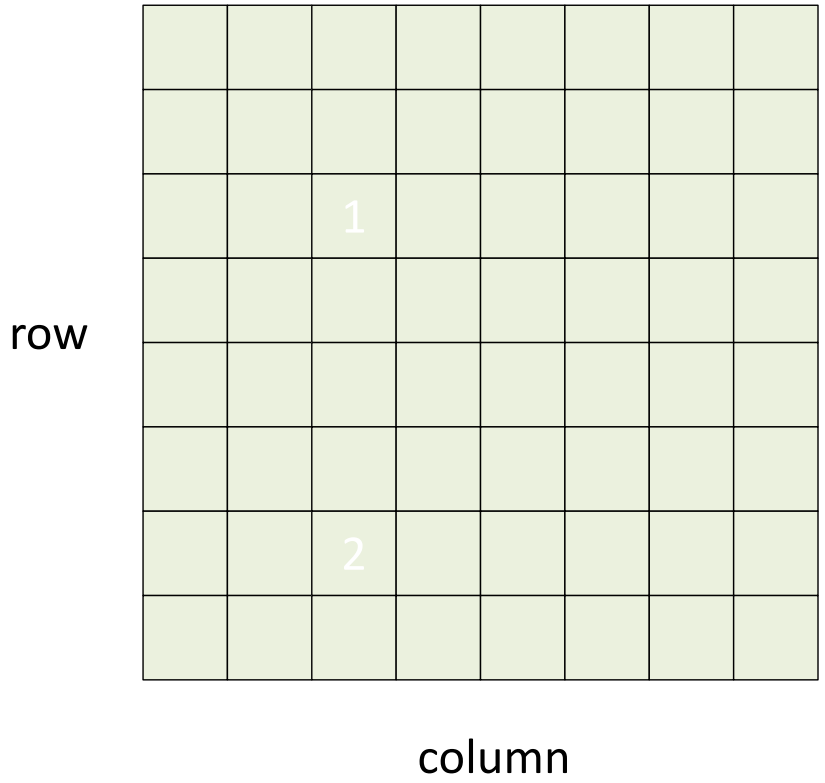
What do errors look like on-chip?

Error Mode
Repeat address
Repeat row
Repeat column
Whole chip
Single Event



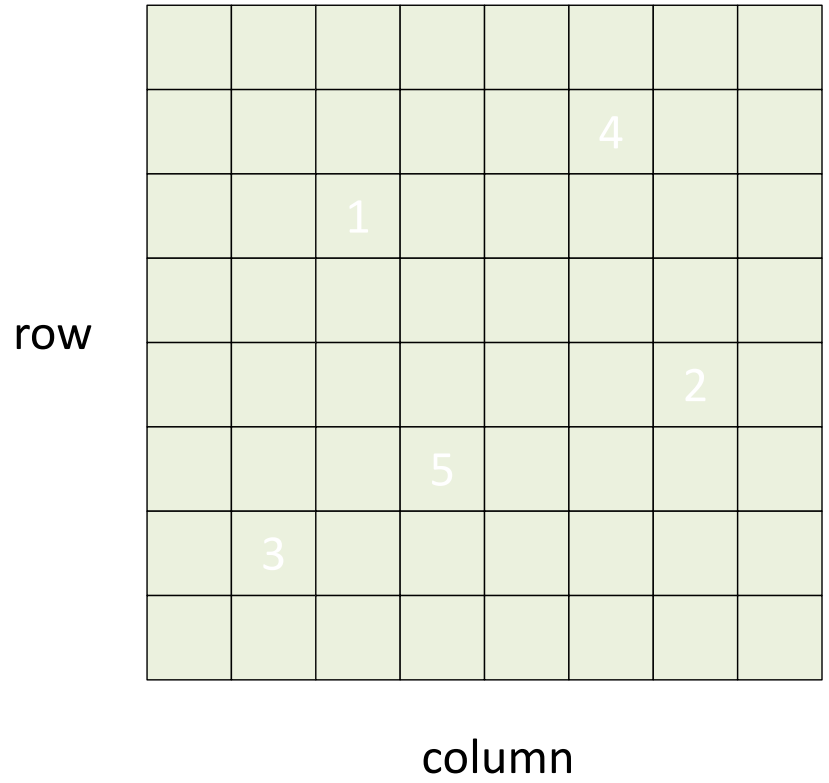
What do errors look like on-chip?

Error Mode
Repeat address
Repeat row
Repeat column
Whole chip
Single Event



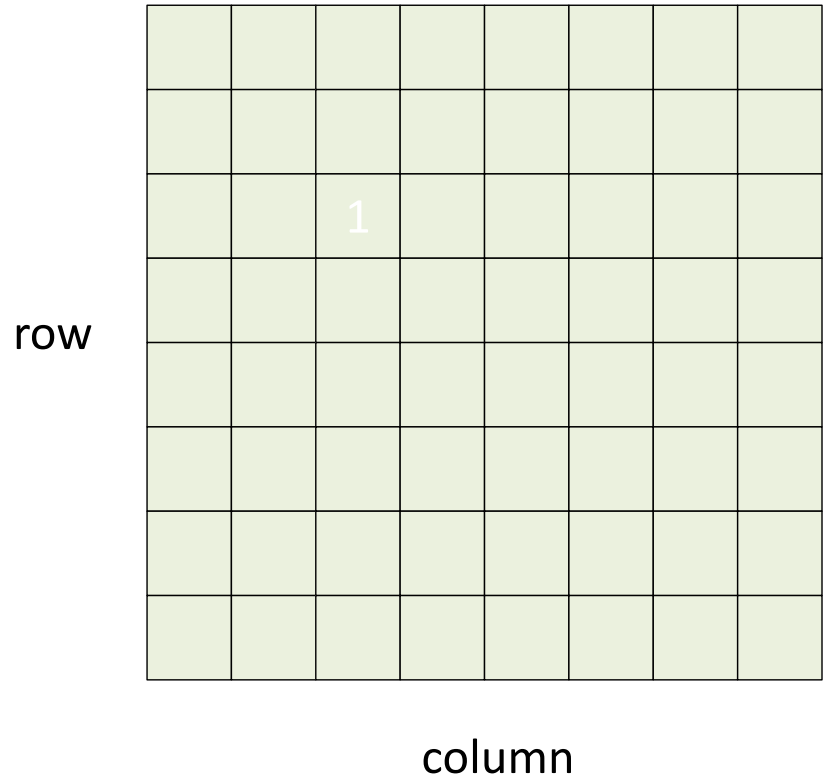
What do errors look like on-chip?

Error Mode
Repeat address
Repeat row
Repeat column
Whole chip
Single Event



What do errors look like on-chip?

Error Mode
Repeat address
Repeat row
Repeat column
Whole chip
Single Event

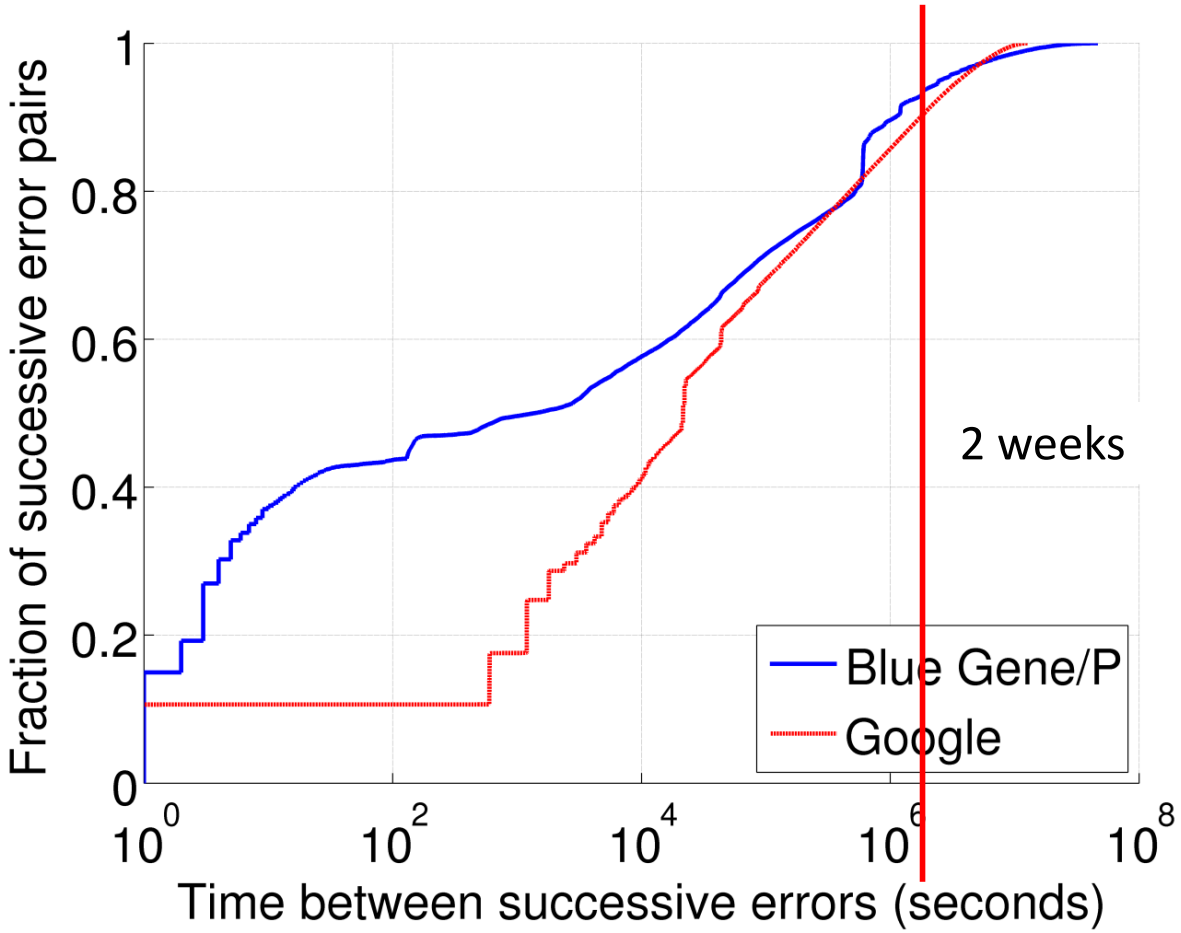


What do errors look like on-chip?

Error Mode
Repeat address
Repeat row
Repeat column
Whole chip
Single Event

- The patterns on the majority of banks can be linked to hard errors.

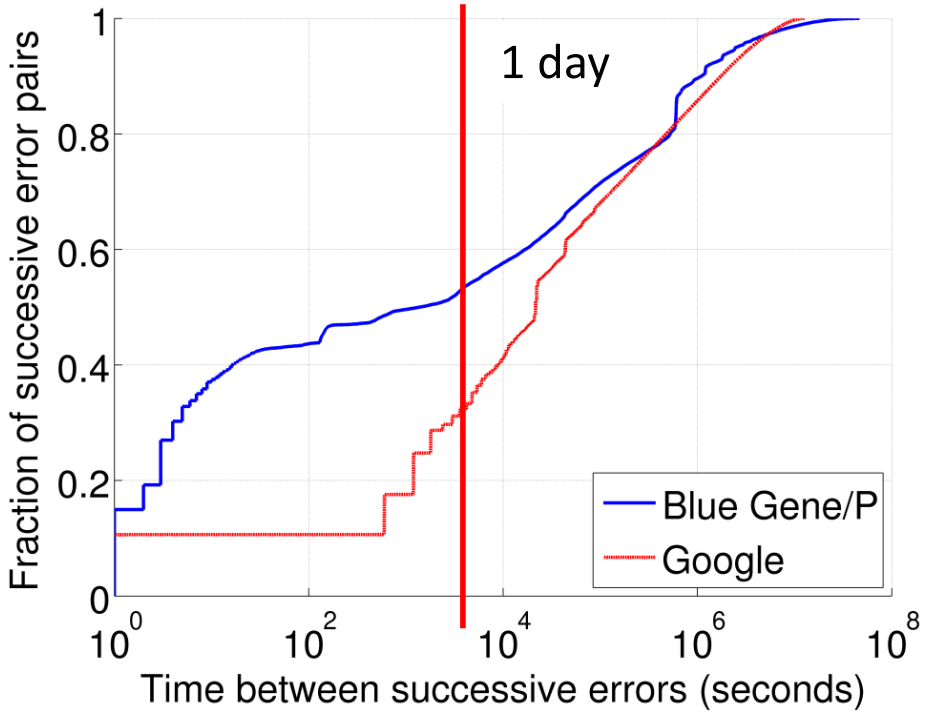
What is the time between repeat errors?



- Repeat errors happen quickly
  - 90% of errors manifest themselves within less than 2 weeks

When are errors detected?

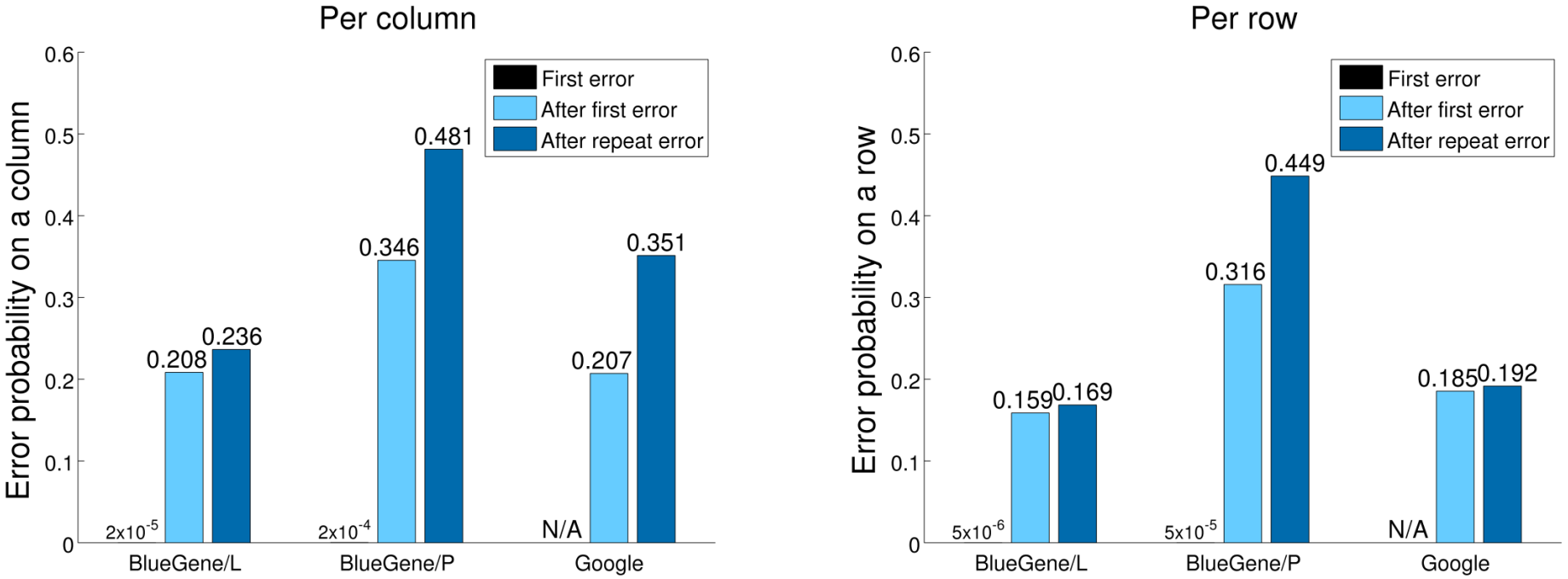
- Error detection
  - Program [read] access
  - Hardware memory scrubber: Google only



- Hardware scrubbers may not shorten the time until a repeat error is detected

How does memory degrade?

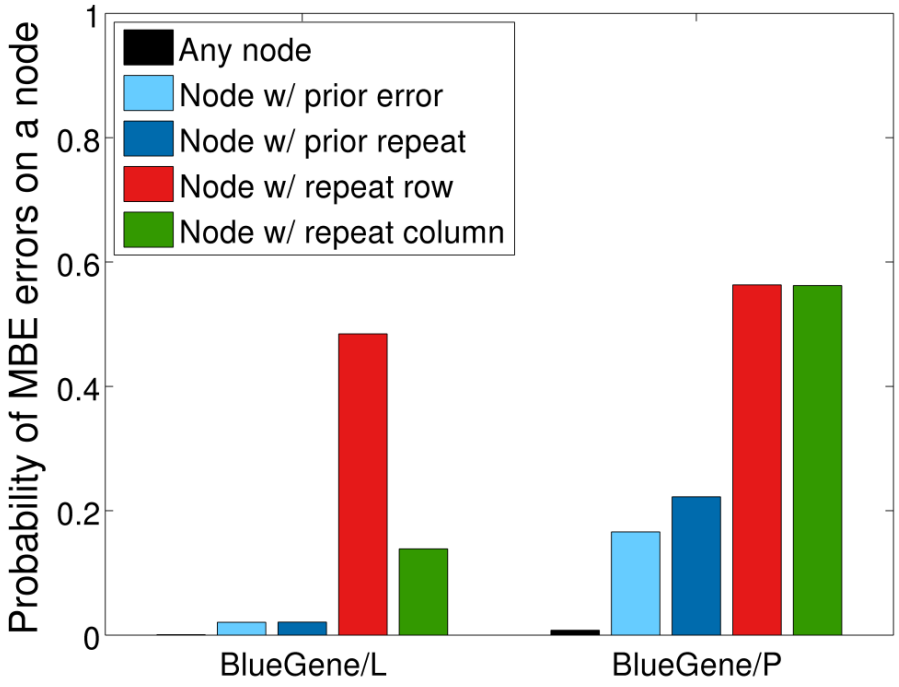
- 1/3 – 1/2 of error addresses develop additional errors
  - Top 5-10% develop a large number of repeats

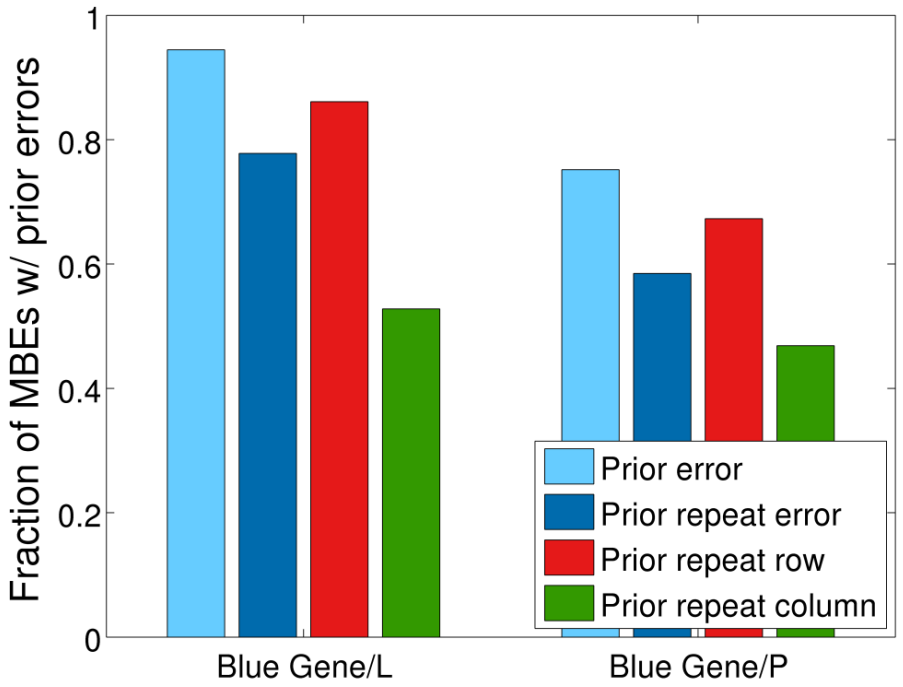


- 3-4 **orders of magnitude** increase in probability once an error occurs, and even greater increase after repeat errors.
- For both columns and rows

How do multi-bit errors impact the system?

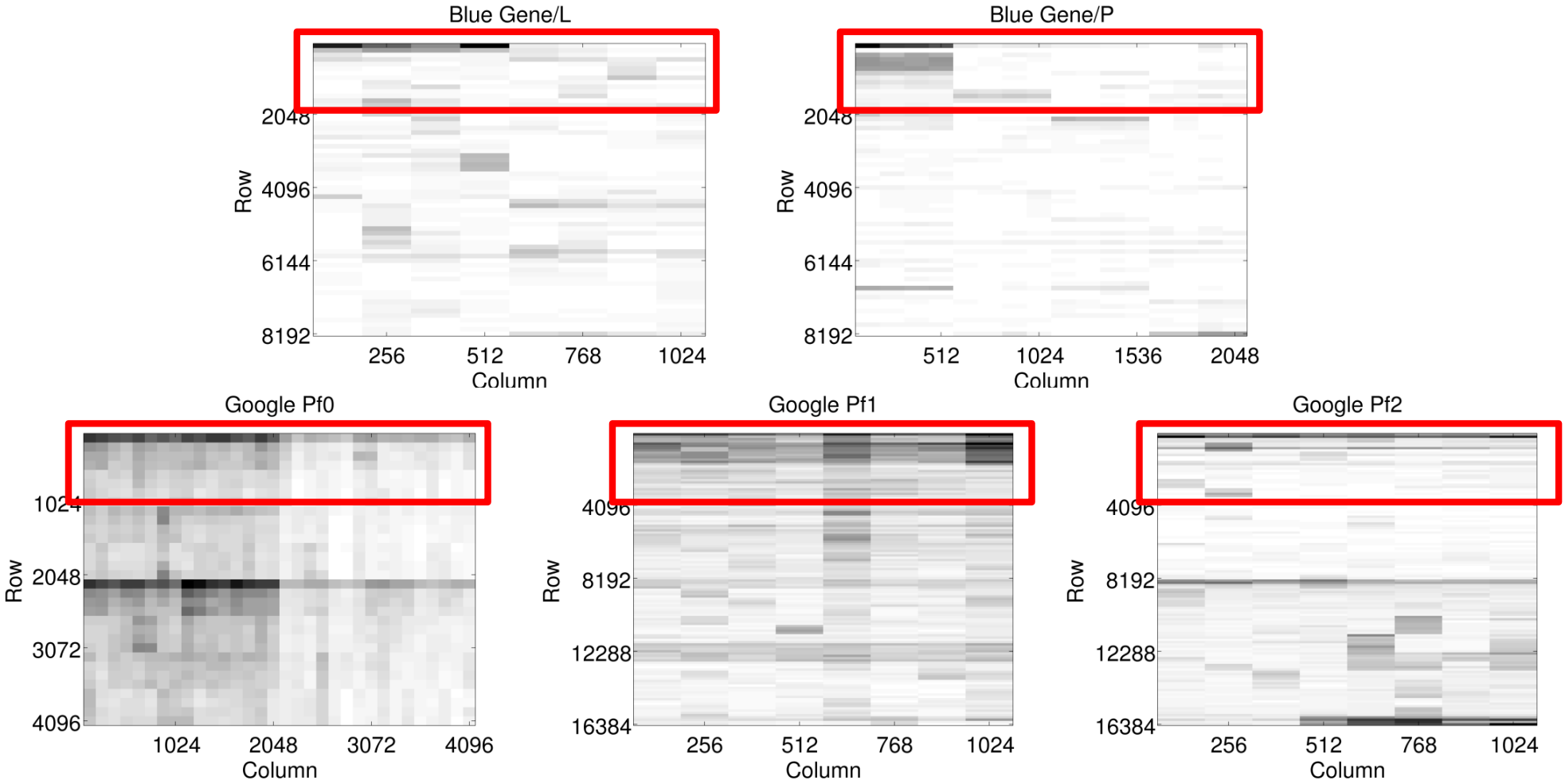
- In the absence of sufficiently powerful ECC, multi-bit errors can cause data corruption / machine crash.
- Can we predict multi-bit errors?





- > 100-fold increase in MBE probability after repeat errors
- 50-90% of MBEs had prior warning

Are some areas of a bank more likely to fail?



- Errors are not uniformly distributed
- Some patterns are consistent across systems
  - Lower rows have higher error probabilities

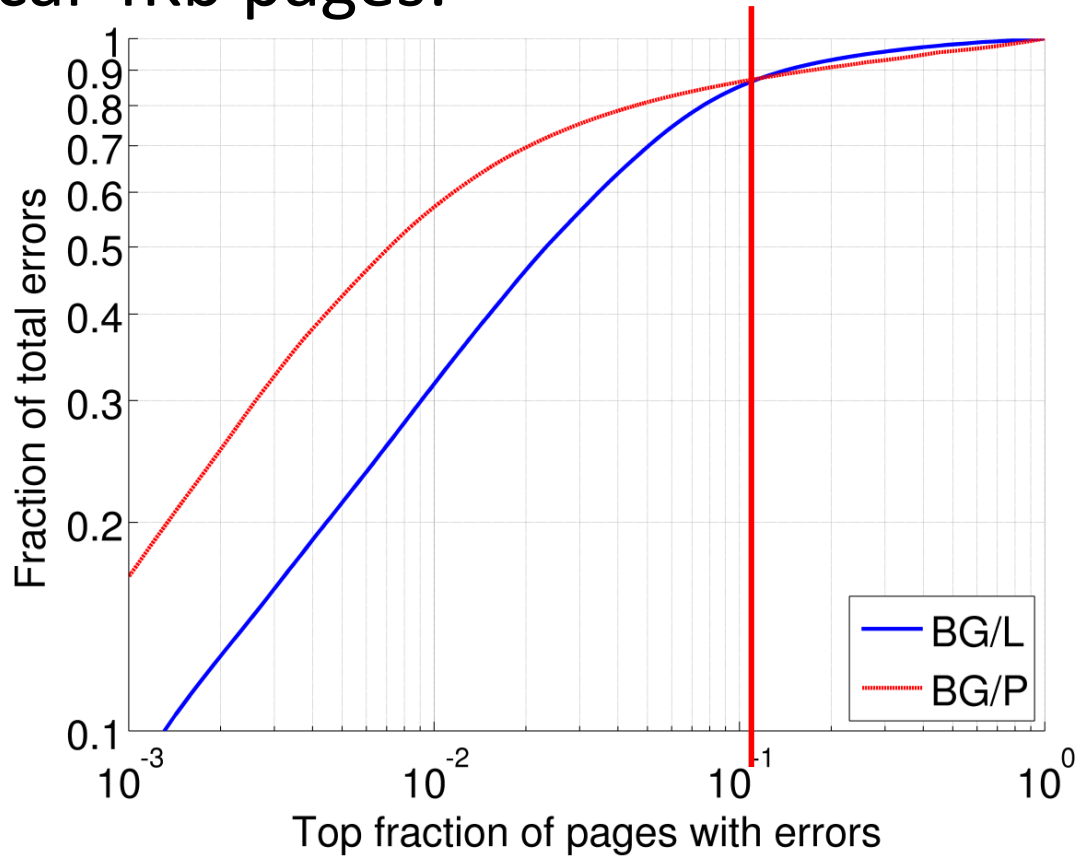
Summary so far

- Similar error behavior across ~300TB-years of DRAM from different types of systems
- Strong correlations (in space and time) exist between errors
- On-chip errors patterns confirm hard errors as dominating failure mode
- Early errors are highly indicative warning signs for future problems

- What does this all mean?

What do errors look like from the OS' p.o.v.?

- For typical 4Kb pages:



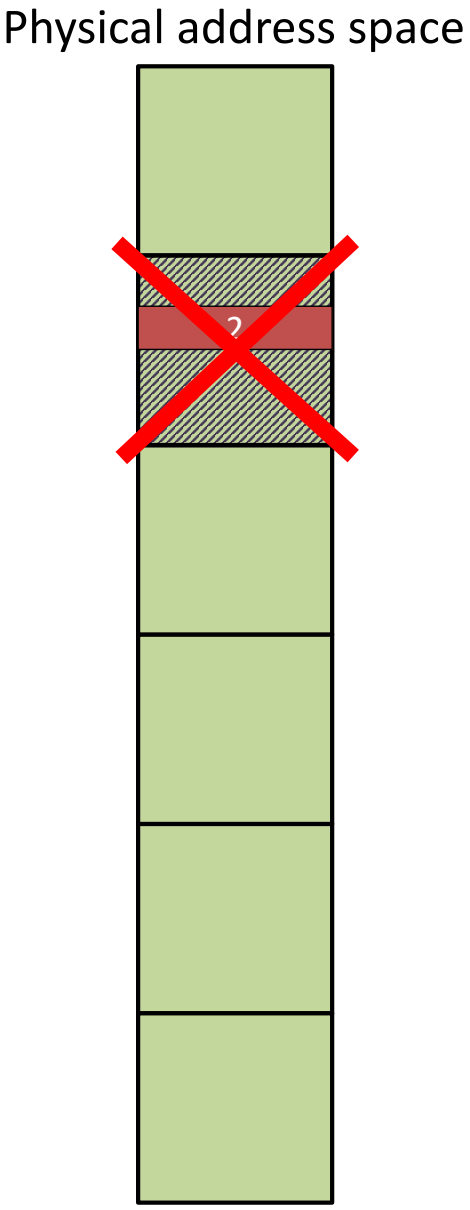
- Errors are highly localized on a small number of pages
  - ~85% of errors in the system are localized on 10% of pages impacted with errors

Can we retire pages containing errors?

- Page Retirement
  - Move page's contents to different page and mark it as *bad* to prevent future use
- Some page retirement mechanisms exist
  - Solaris
  - BadRAM patch for Linux
  - But rarely used in practice
- No page retirement *policy* evaluation on realistic error traces

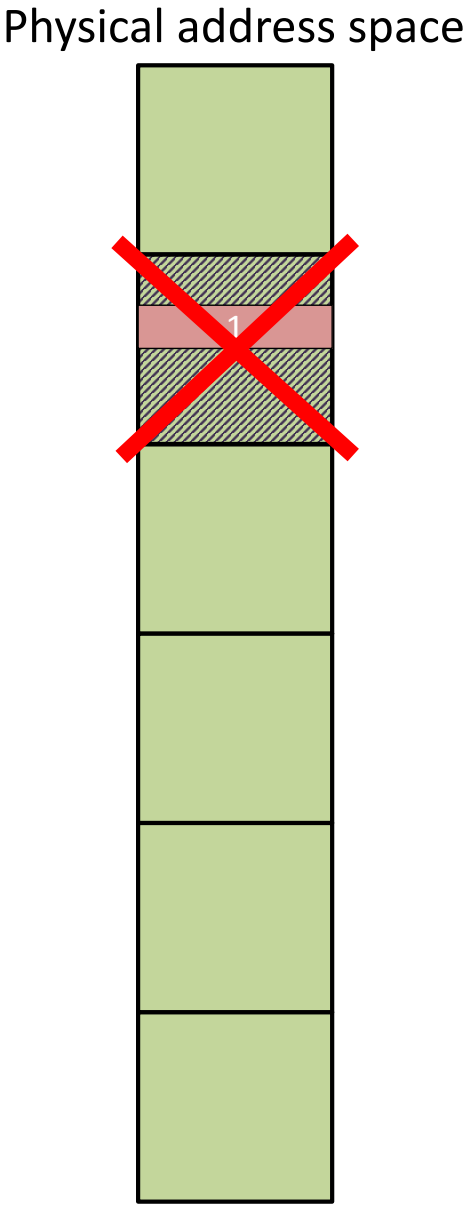
What sorts of policies should we use?

- Retirement policies:
  - Repeat-on-address
  - 1-error-on-page
  - 2-errors-on-page
  - Repeat-on-row
  - Repeat-on-column



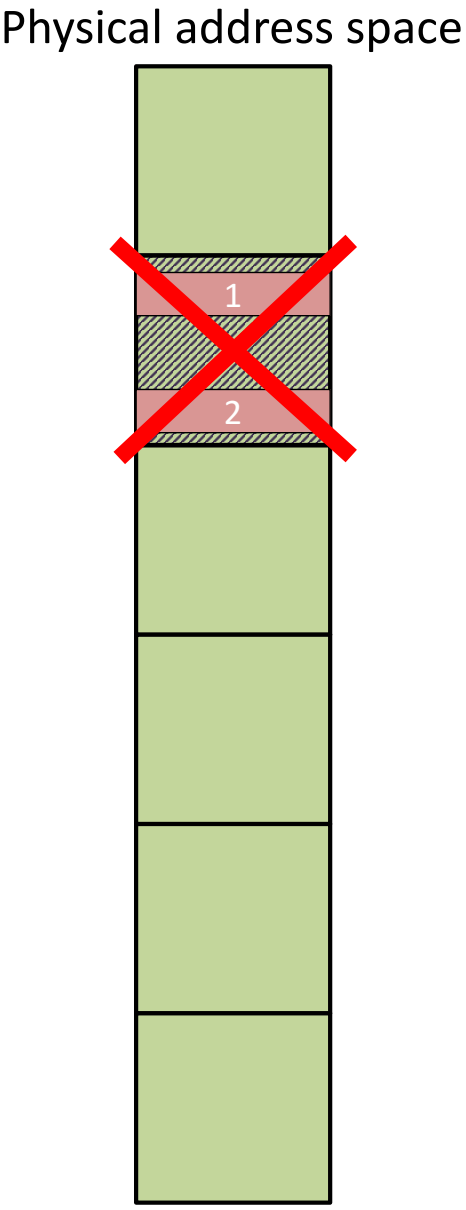
What sorts of policies should we use?

- Retirement policies:
  - Repeat-on-address
  - 1-error-on-page
  - 2-errors-on-page
  - Repeat-on-row
  - Repeat-on-column



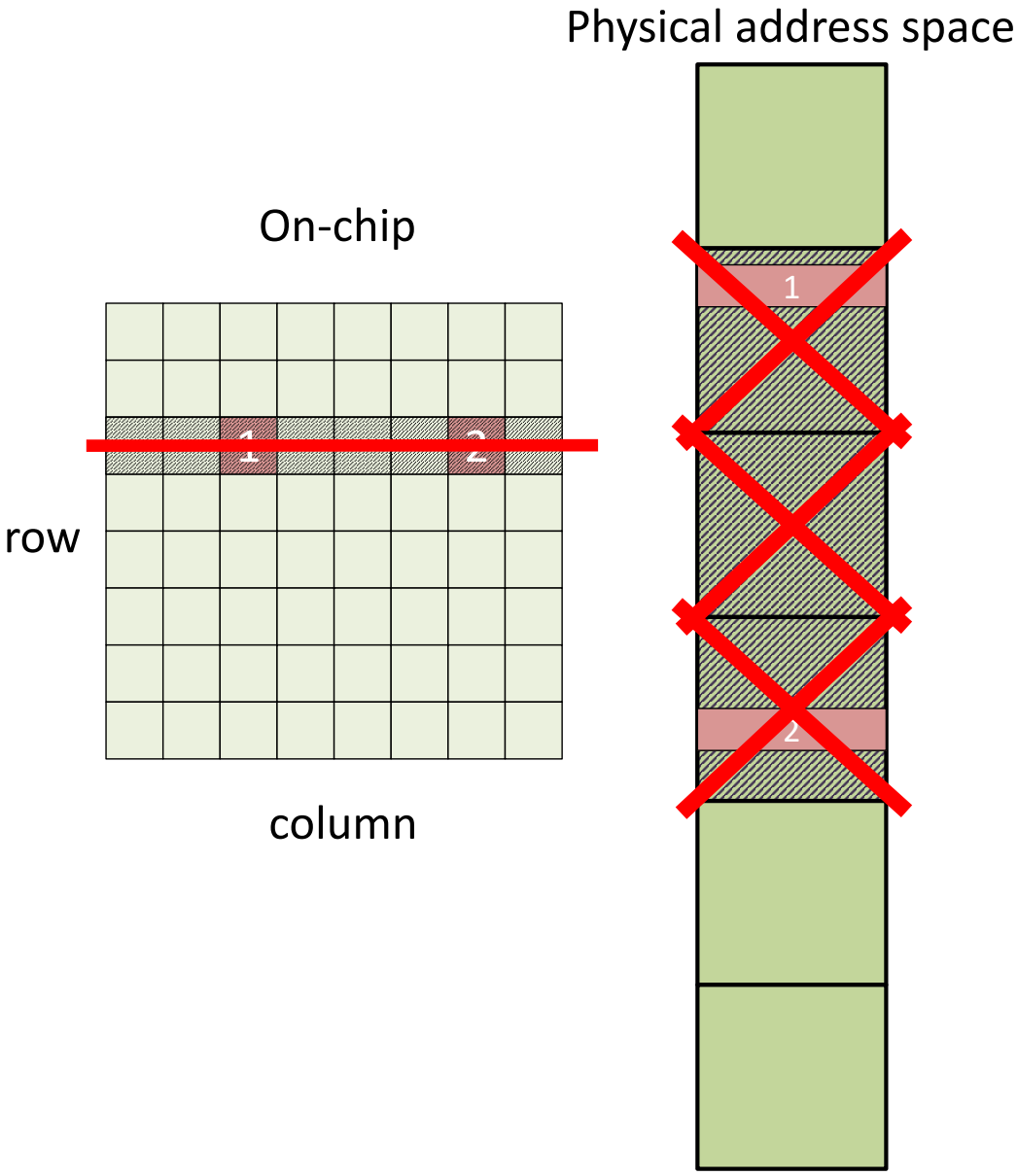
What sorts of policies should we use?

- Retirement policies:
  - Repeat-on-address
  - 1-error-on-page
  - 2-errors-on-page
  - Repeat-on-row
  - Repeat-on-column



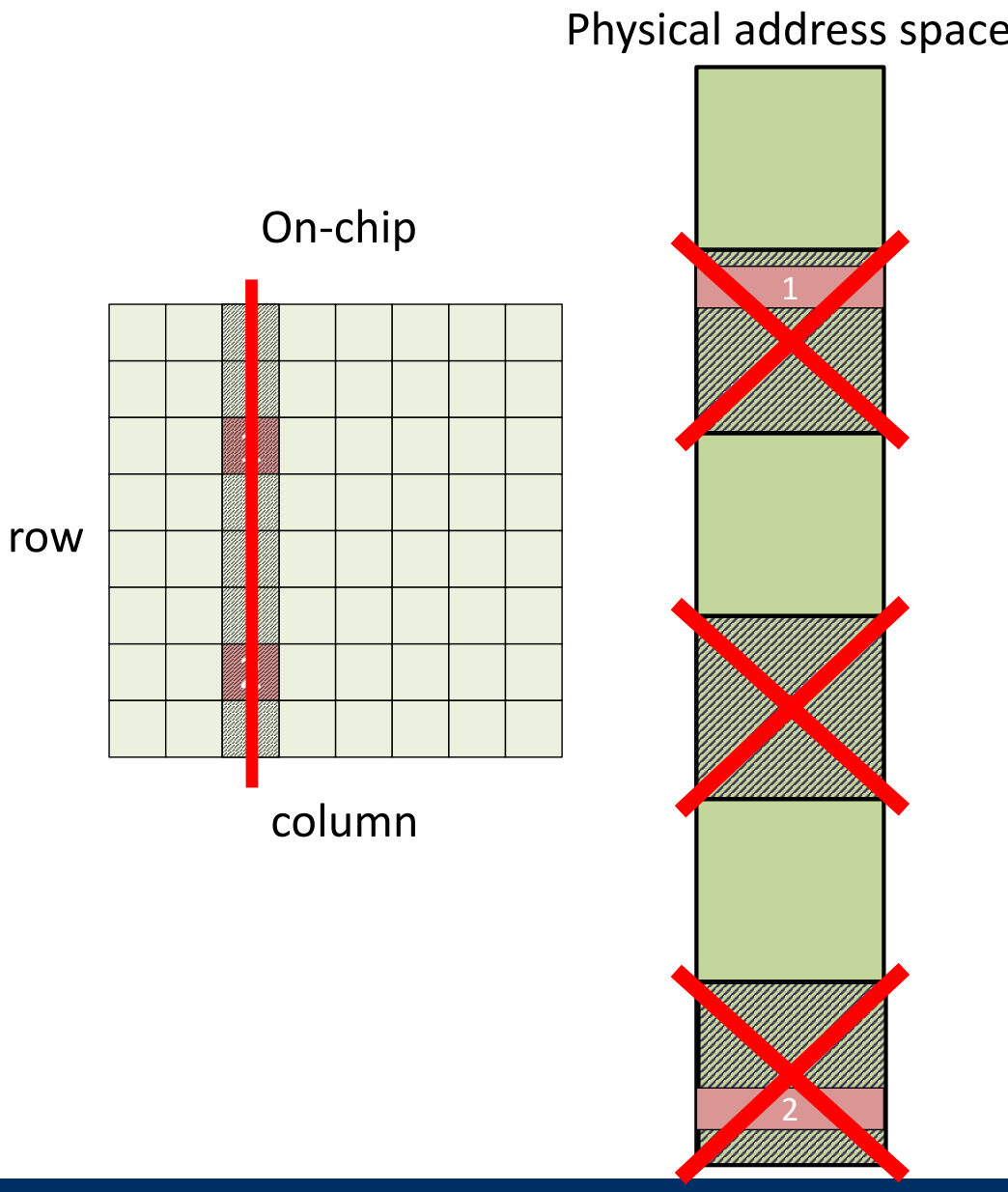
What sorts of policies should we use?

- Retirement policies:
  - Repeat-on-address
  - 1-error-on-page
  - 2-errors-on-page
  - Repeat-on-row
  - Repeat-on-column



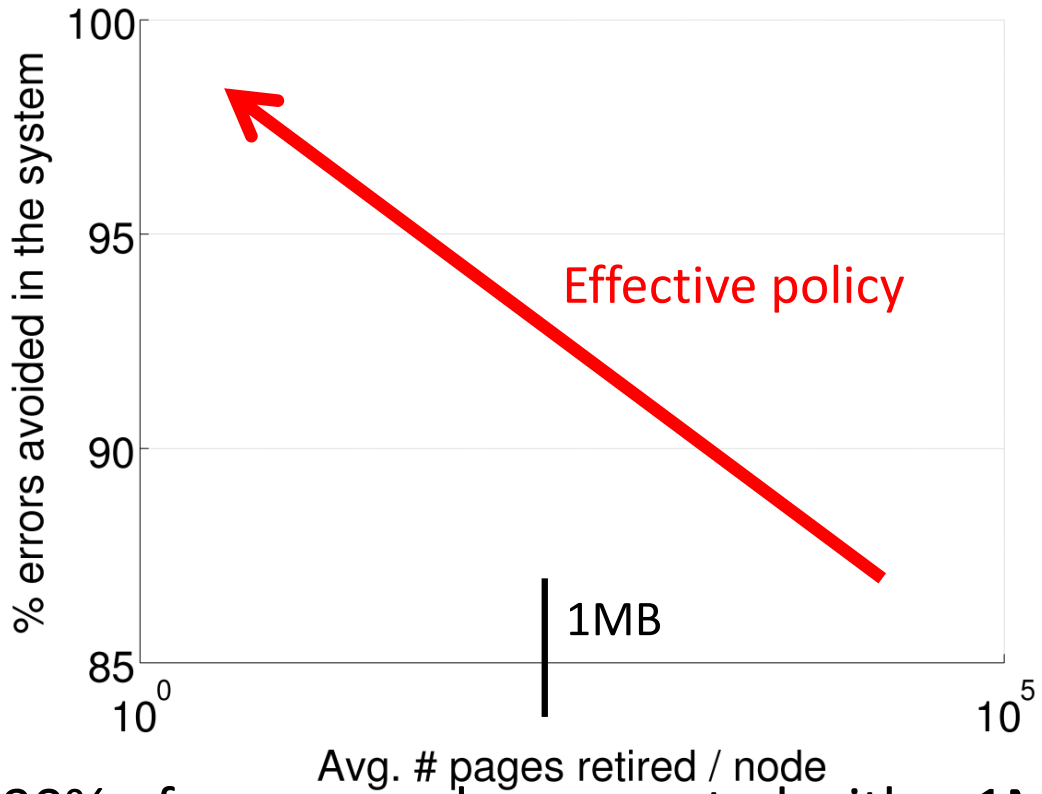
What sorts of policies should we use?

- Retirement policies:
  - Repeat-on-address
  - 1-error-on-page
  - 2-errors-on-page
  - Repeat-on-row
  - Repeat-on-column



How effective is page retirement?

- For typical 4Kb pages: BlueGene/L (MBE)



- More than 90% of errors can be prevented with < 1MB sacrificed per node
  - Similar for multi-bit errors

Implications for future system design

- OS-level page retirement can be highly effective
- Different areas on chip are more susceptible to errors than others
  - Selective error protection
- Potential for error prediction based on early warning signs
- Memory scrubbers may not be effective in practice
  - Using server idle time to run memory tests (eg: memtest86)
- Realistic DRAM error process needs to be incorporated into future system design

Higher-Level Take-Aways

- Reasoning about large-scale system behaviour is hard
  - This includes failures/reliability
- Understanding current production systems is crucial to building next generation of systems.
- Solutions sometimes require cooperation between hardware and software