BLACK SHEEP → BLACK SWAN

"An idiom used to describe an odd or disreputable member of a group, especially within a family. The term has typically been given negative implications, implying waywardness." - Wikipedia

"A metaphor that encapsulates the concept that the event is a surprise (to the observer) and has a major impact. After the fact, the event is rationalized by hindsight." - Wikipedia

Resilience: From Black Sheep to Black Swan

Nathan DeBardeleben, PhD
Los Alamos National Laboratory
High Performance Computing
Ultra Scale Research Center, Resilience Lead

LA-UR-12-20261

Los Alamos NATIONAL LABORATORY EST. 1943

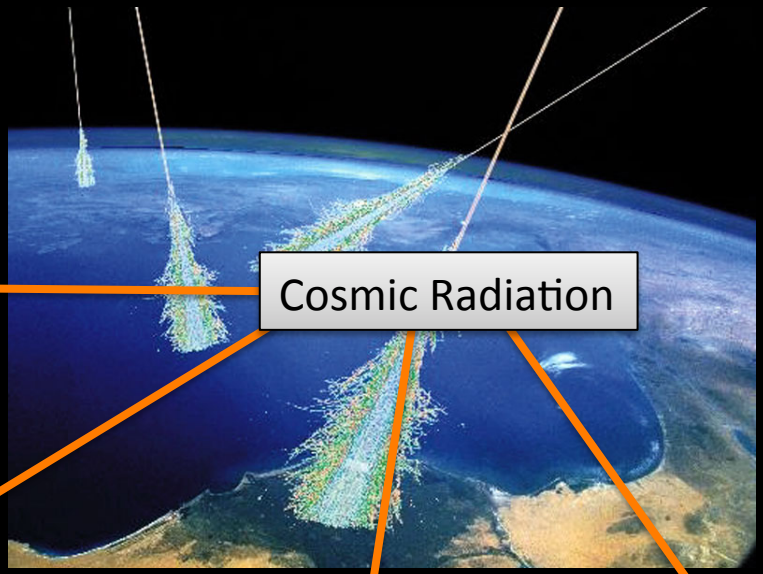
Today's Journey

- Near Term Resilience:
  - Challenges
  - Opportunities
  - Solutions
- Big memory:
  - Silent data corruption
  - Testing
    - Neutron beam
    - Fault injection
- Checkpoint path changes
  - Models
  - Top-down design for resiliency
- Production-centric resilience

We Like Big Memory!

- By square meters, memory has much greater exposure to radiation than processors
- If we're going to talk about memory, we need to talk about single event upsets (SEUs), cosmic radiation, voltage droops, temperature fluctuations, etc.

Particularly interesting. Let's focus on this!



Bit Flips

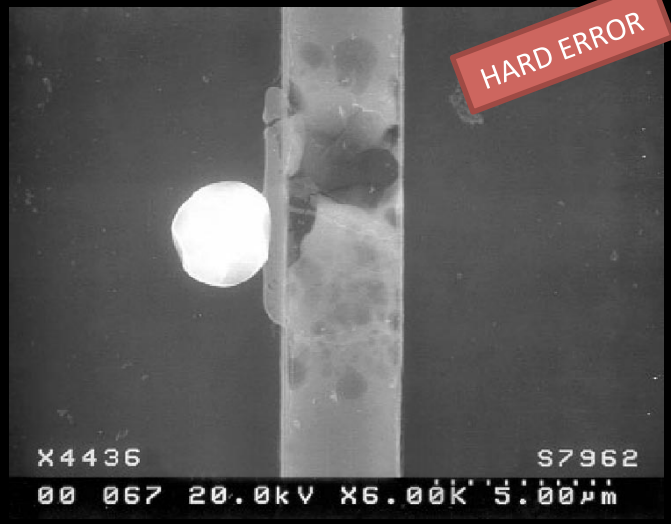
SOFT ERROR

1011101 → 1001101

Single bit flips are corrected by ECC, double bit flips are detected, more are uncaught

Stuck Bits

HARD ERROR



Damage caused by single event latchup

Becker, H.N.; Miyahira, T.F.; Johnston, A.H.; , "Latent damage in CMOS devices from single-event latchup," Nuclear Science, IEEE Transactions on , vol.49, no.6, pp. 3009- 3015, Dec 2002

Burst Errors

SOFT ERROR

Many bit flips on an entire page of memory. Not caught by ECC. Attributed to memory logic (refresh / control)

Stuck Pages

Largely an unknown

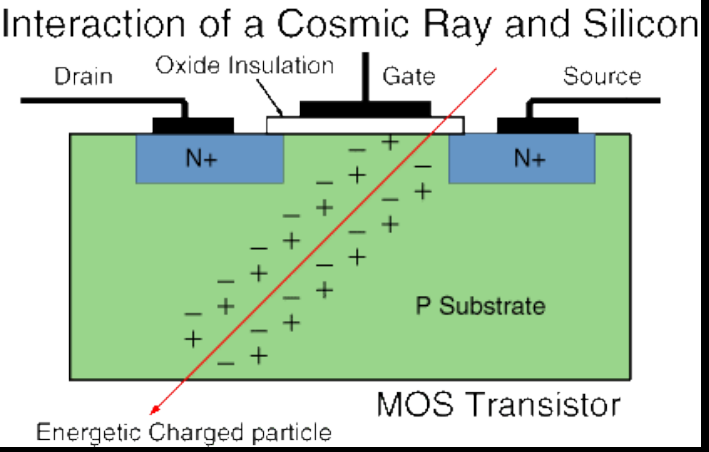
HARD ERROR

This is a picture of a SEL (latchup) – not a stuck bit – but the effect should be similar

Bishop George Berkeley (1685 – 1753)

"esse est percipi" ("to be is to be perceived")

- Bit flips in unused memory do not:
  - Cause silent data corruption
  - Crash your system
  - Get caught by ECC
- Even if it's in a data region, if you never read from it again you will not see the bit flip
- This makes it really difficult to study expected rates of memory problems because it's highly correlated to your:
  - Application (and its memory access patterns)
  - Memory (manufacturer, model, DDRX, number of chips, etc)
  - Neutron flux at location
  - Final factor in SDC = system size (neutrons * system size is what worries me about large memory systems)
- As with most other areas of resilience – this topic is extremely cross-cutting!
- We are studying this but:
  - The hardware is always changing (Quinn's tests were of DDR2 and of very small DIMMs : "we will fly DDR2 in 2018") – can we really use this data?
  - The applications vary
  - Making extrapolations is hard!



Aerospace Corporation

Do We Need Better ECC?

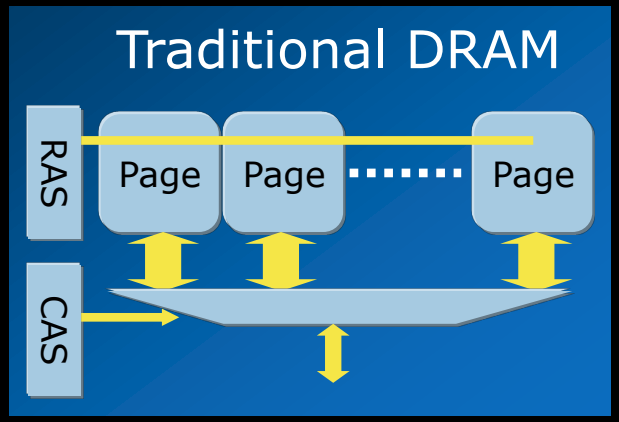
- Short answer is no
- Due to the way logical memory is physically distributed on a chip, logical bits are not generally susceptible to multi-bit error
- Indirectly ionizing neutrons do not have the energy to get very far – they cause damage in a small region
- What we really need to be worried about are "burst errors" in the memory circuitry (not data cells)

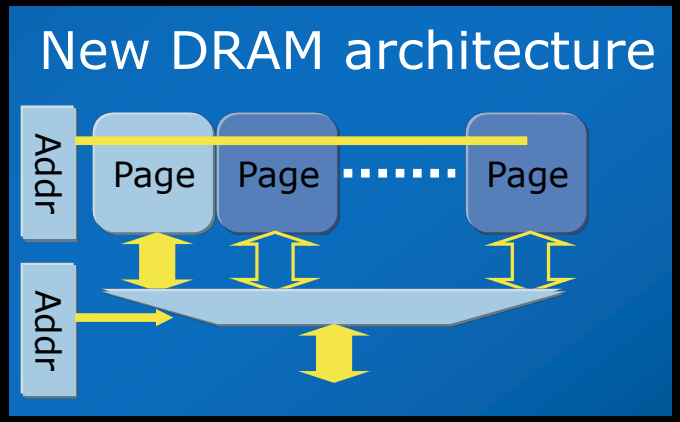
Los Alamos Neutron Science Center (LANSCE)

Quinn, H.; Graham, P.; Fairbanks, T.; , "SEEs Induced by High-Energy Protons and Neutrons in SDRAM," Radiation Effects Data Workshop (REDW), 2011 IEEE , vol., no., pp. 1-5, 25-29 July 2011

Memory Resilience – Getting Helped by Power Requirements

- Well known that DRAM is power hungry, with respect to total system power budget
  - Power constraints are driving us to rethink DRAM "overfetch" where thousands of bit-lines are activated in many DRAM chips only to return a single cache line
    - Aniruddha N. Udipi, Naveen Muralimanohar, Niladrish Chatterjee, Rajeev Balasubramonian, Al Davis, and Norman P. Jouppi. 2010. Rethinking DRAM design and organization for energy-constrained multi-cores. In Proceedings of the 37th annual international symposium on Computer architecture (ISCA '10). ACM, New York, NY, USA, 175-186.
  - Shekhar Borkar (Intel) suggests this might drive power from 150 pJ/bit to 8-10 pJ/bit
- Some of the ways being discussed to restructure DRAM could help resilience! We are hopeful.
- And the server market probably will drive this, so we might actually get it!
- Is this a tipping point where we can influence the memory makers?





Shekhar Borkar (Intel) – *Exascale Challenges, Why Resiliency?* – 2/21/2012

What about memory hard failure?

- SGI's proprietary "memlog" tool:
  - More intelligent logging – "DIMM location, DRAM, rank, bank, row and column address, and failing bit(s)"
  - Keeps a failure history to look for patterns
  - Helps keep from replacing DIMMs too early and at the same time helps identify failing DIMMs more quickly than conventional schemes
  - This exists today and we use this on some systems and see a lower DIMM replacement rate
    - Is it the tool or is the memory better? Unknown.
  - Part of this is because the memory is very well characterized
  - Perhaps this needs to be a value-add in procurements
- Where to take this next?

Memory Over-Provisioning

- Technique:
  - Reserve spare bits
  - Look for error patterns
  - Map out failing *portions* of hardware
  - Map in reserved spares
  - Hardware has some predictable lifetime (~5 years) before all spare regions are gone
  - This is fail in place
- We do this today on JBODs and SSDs/Flash
- Might be time for this approach in memory
  - Everything old is new again!
- And we can get it now – Tezzaron Semiconductor Bi-STAR Technology
- Helps with wear but also yield
  - Perhaps the yield angle will be how vendors get interested in this and we will benefit

What About 3D Memory?

- Probably won't see it in the Trinity timeframe, but certainly interesting for exascale
- Soft error perspective
  - Probably don't need to worry about the organization (vertical vs. horizontal)
    - Indirectly ionizing neutrons:
      - 50MeV -> Silicon or Magnesium heavy ion
      - 300MeV -> Lithium or Carbon heavy ion
    - Either way, the heavy ion does not have the range to get to another device
  - The middle layers of a 3d die could be hotter and therefore have higher neutron sensitivities
- Hard fail perspective
  - All the more need for mapping out of bad regions as the cost to replace entire 3D units will likely be high
- 3D memory should also provide more error correcting capability

Memory Resilience – We Might Make It!

- Soft Errors:
  - While trends are working against us due to our scale, DRAM redesign for power could have a side effect on improving memory resiliency to soft errors and SDC
- Hard Fails:
  - Techniques exist today which make this more intelligent
  - Memory Overprovisioning:
    - And we might even get to use them if we can make the case that they increase yield and vendors want to do this for the consumer (or server) market
- We need to stay engaged and make sure these technologies mature in a way that benefits our applications
- CPUs, networks, storage systems, these remain to be addressed at a later time

Using Fault Injection to Identify Vulnerable Application Regions

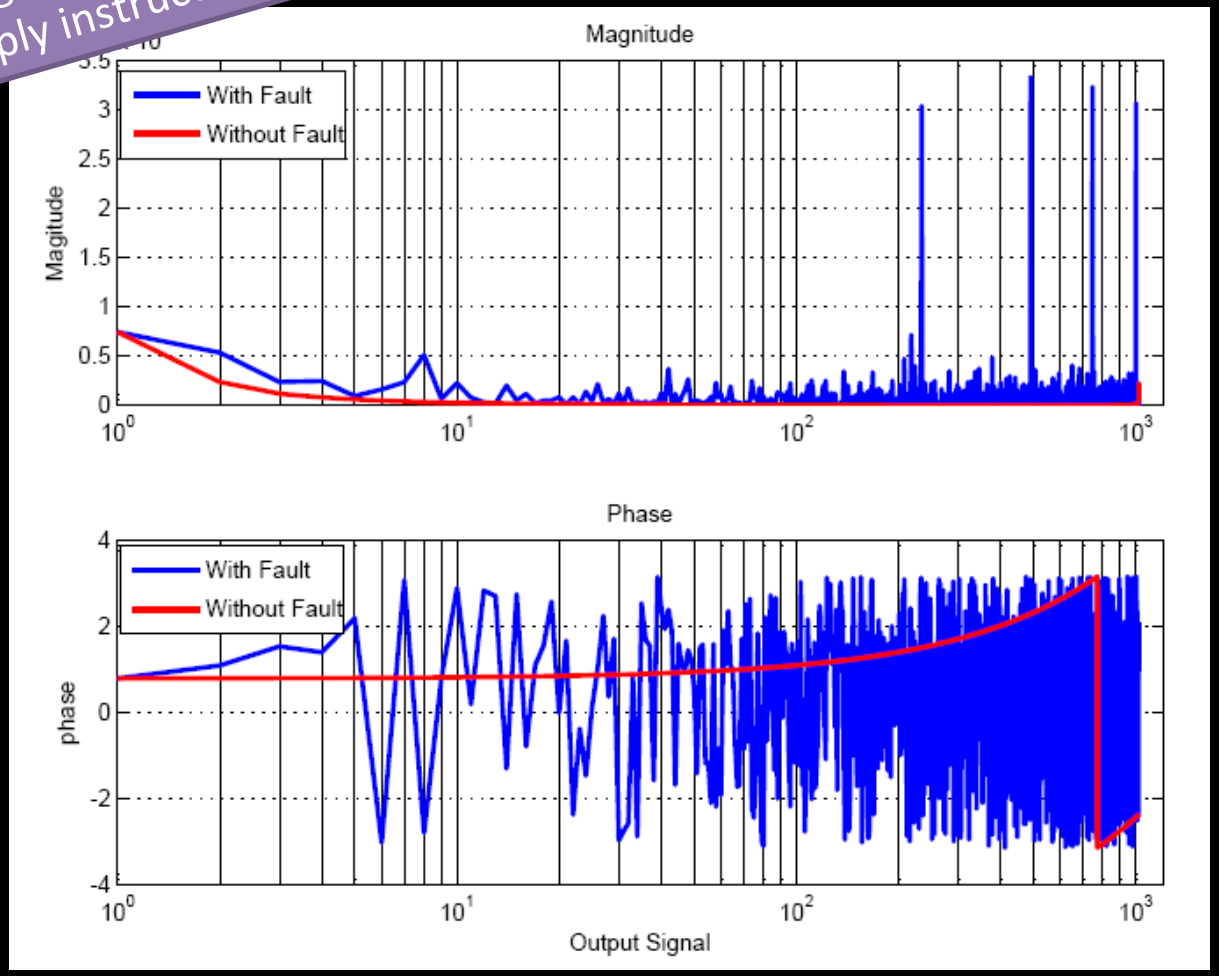
- Purpose: To better understand, empirically, how our applications perform on real hardware so as to identify their susceptibilities to silent data corruption and use this to motivate research into partial redundancy while at the same time testing resiliency mechanisms.
- Hasn't fault injection been done before?
  - Yes. Lots! In hardware and software
  - Much of the work focuses on the probability a random SDC will affect an application.
  - Our take is *ever so slightly* different: Target SDC and observe how the application responds
    - Move past the "tree falls in the forest, if no one was there, did it make a sound" type of random data corruption to see if we can classify applications by their fault characteristics
    - Target machines at specific applications based on their fault characteristics
- Why go through all this SDC protection if some applications don't need it?
- What if we could use special less (or more) reliable systems just for those apps?
- What if the application could dynamically change its protection?

Soft Error Fault Injection (SEFI) Framework

- Modify instructions dynamically – cause them to produce the wrong answer
  - Controllable error
  - Controllable probability
- Initial version used processor emulator VM (QEMU)
  - Nathan DeBardeleben, et. al Experimental Framework for Injecting Logic Errors in a Virtual Machine to Profile Applications for Soft Error, Resilience Proc. of Resilience, the 17th International European Conference on Parallel and Distributed Computing (Euro-Par), September 2011.
- New approach uses PIN dynamic instrumentation
- Literally changes opcodes "on the fly"

FFT – With and Without Fault Injection

This was a single random bit flip in a multiply instruction!



. . . Or maybe not?

- We talk about how DOE is scared of SDC, but are we being saved by the way we do business?
- The statistical nature of our work helps us here!
- Parameter studies / sweeps are the norm
- If only more of our applications were Monte Carlo style we'd be better off!
- But are we not doing MC at the job level (instead of at the process level)?

Are We Tilting at Windmills?

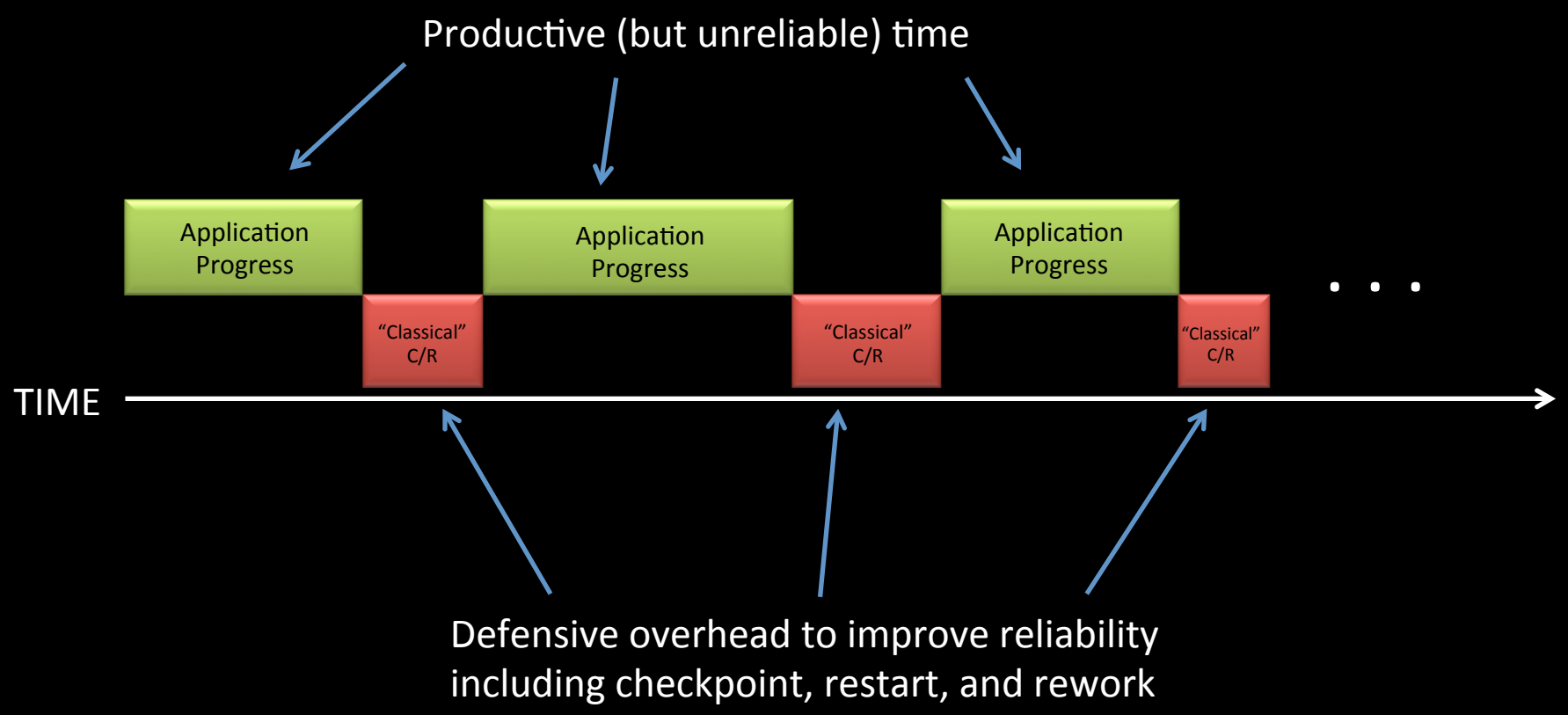
I don't know.

"But I try not to think with my gut. Really, it's okay to reserve judgment until the evidence is in." - Carl Sagan, The Burden Of Skepticism, The Skeptical Inquirer, Vol. 12, Fall 87

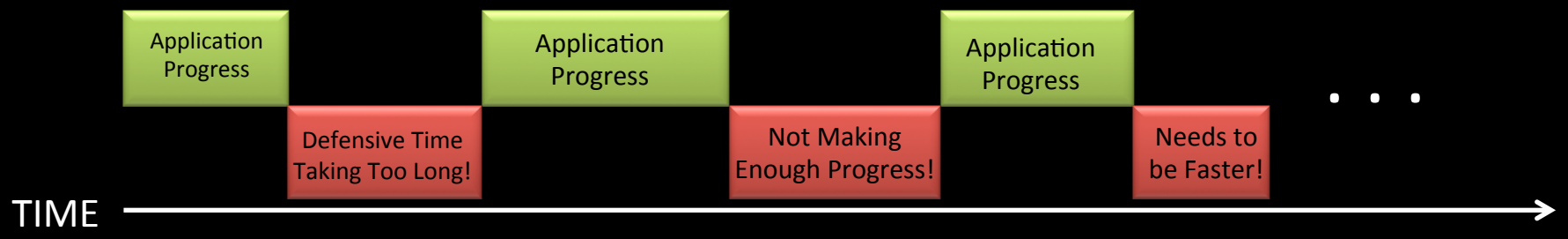
We need to test (our hardware, our applications)

. . . let's move on from cosmic radiation effects to . . .

We are familiar with basic checkpoint / restart . . .

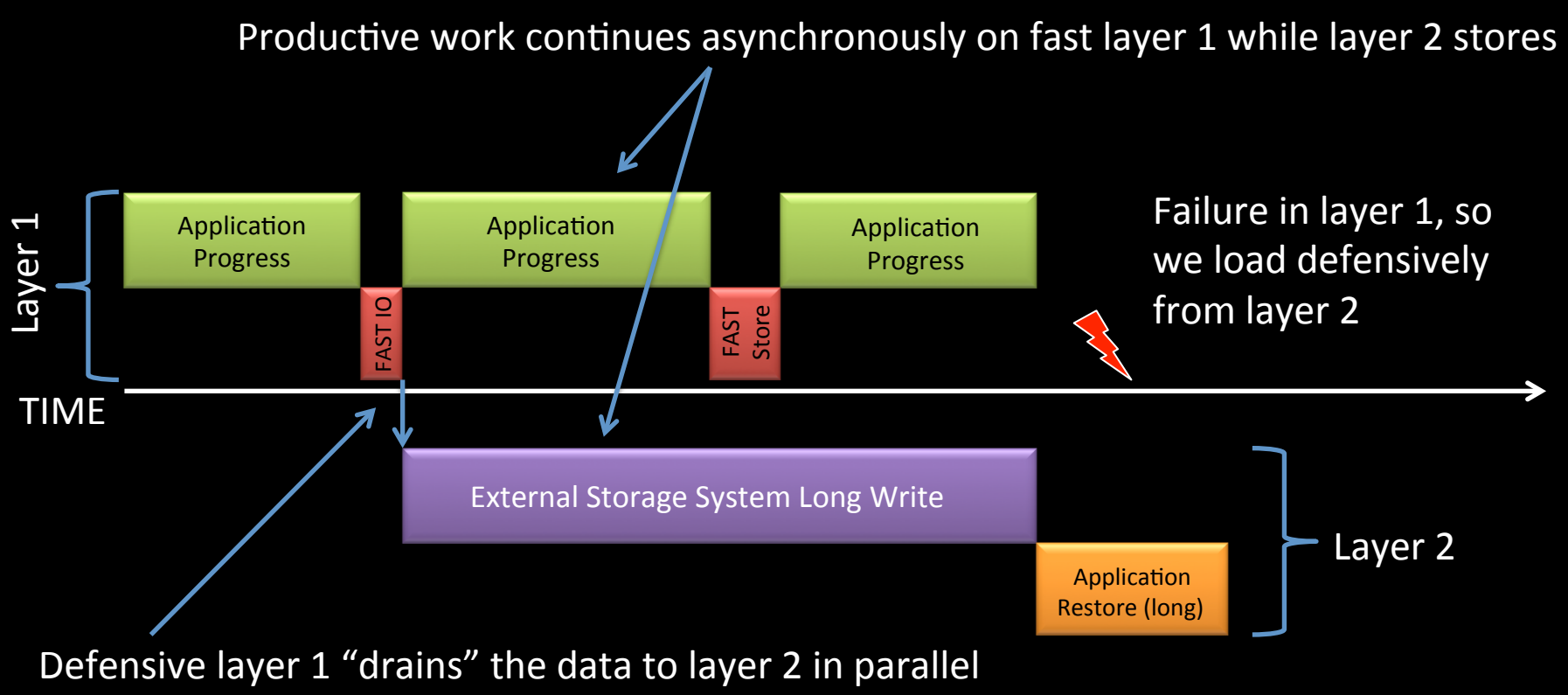


. . . But what if the defensive time grows to a large enough portion of total system time?



We have indications that this is happening and, unchecked, defensive time will continue to grow to larger and larger portions of total system time.

One approach to deal with this is to add a new layer



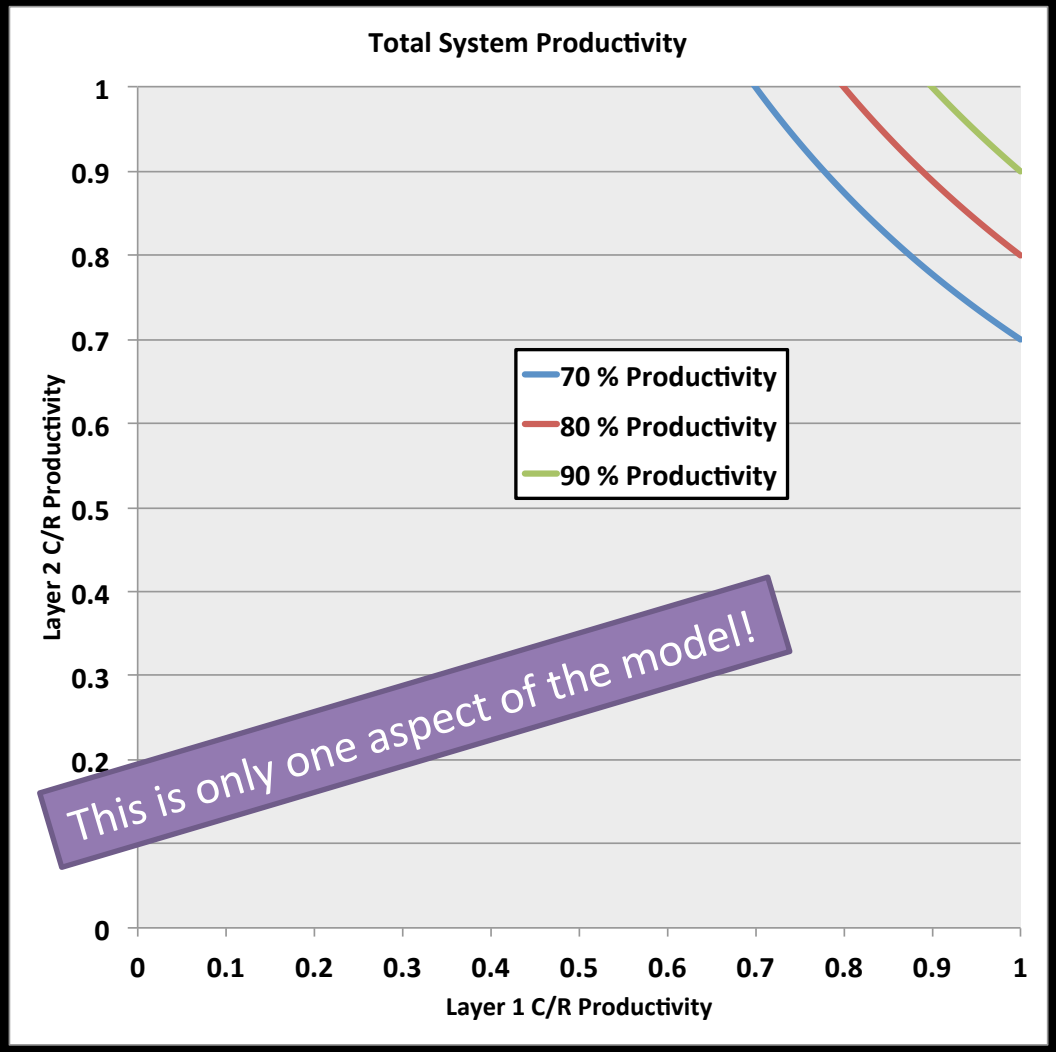
If the fast layer 1 is somewhat reliable, frequency of draining to layer 2 can be reduced.
Layer 1 partially mitigates low MTTI of the platform.
Layer 2 mitigates problems with layer 1.

Burst Buffer

- . . . And there are many other approaches to how this could work
- Generally, we term this technology a "burst buffer"
- Expected to be needed beyond petascale
- This affects the fundamental mathematical models used in resilience
- We are working on this:
  - Complicated
  - Driven by speeds and feeds, individual layer reliabilities, application checkpoint sizes, application quiescence intervals, recovery times, etc
  - **But it drives a top-down design for resiliency**

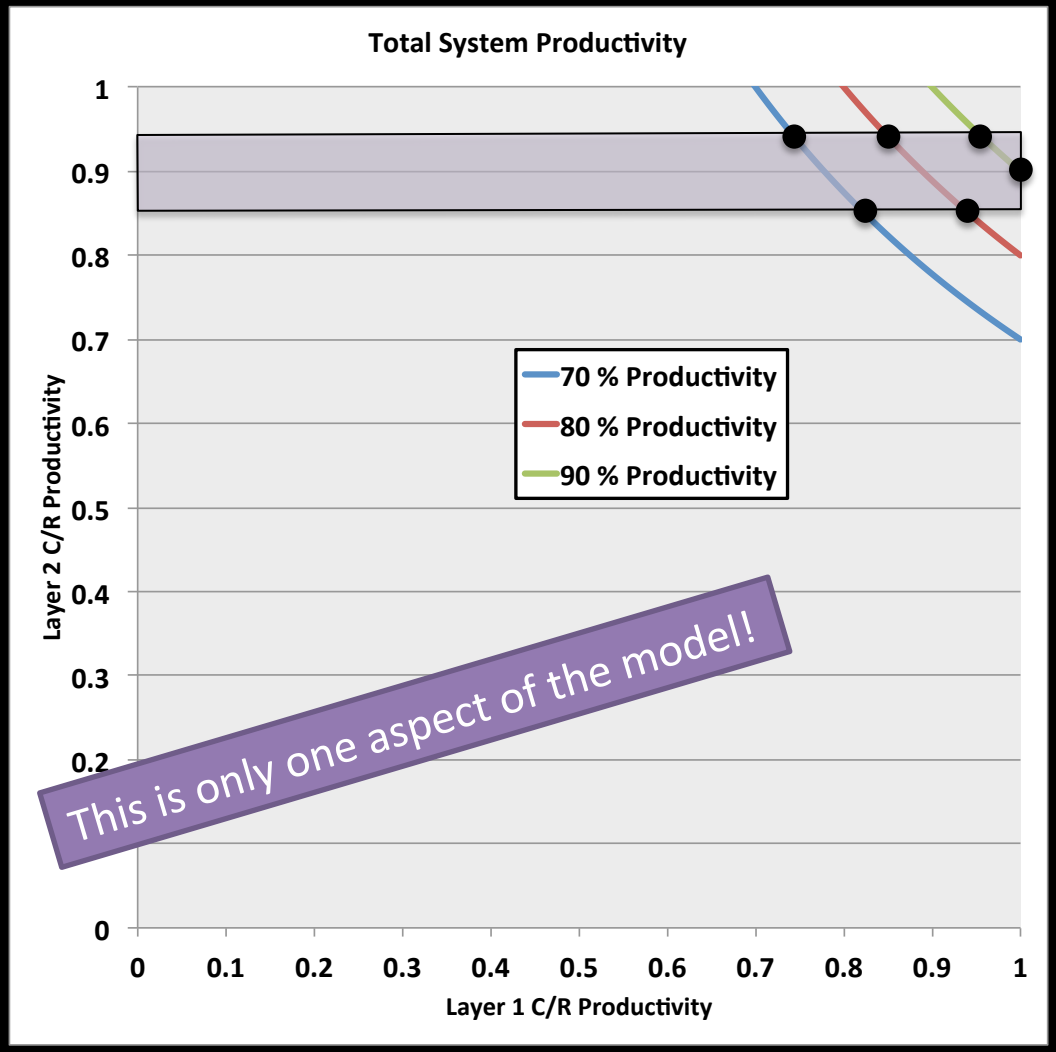
Multi-layer Checkpoint / Restart Modeling

- Daly's models built upon Young's models
- Further improvements on Daly's models have been done to include checkpoint to medium A or medium B
- Our work is novel as it introduces the notion of asynchronous progress in the layers



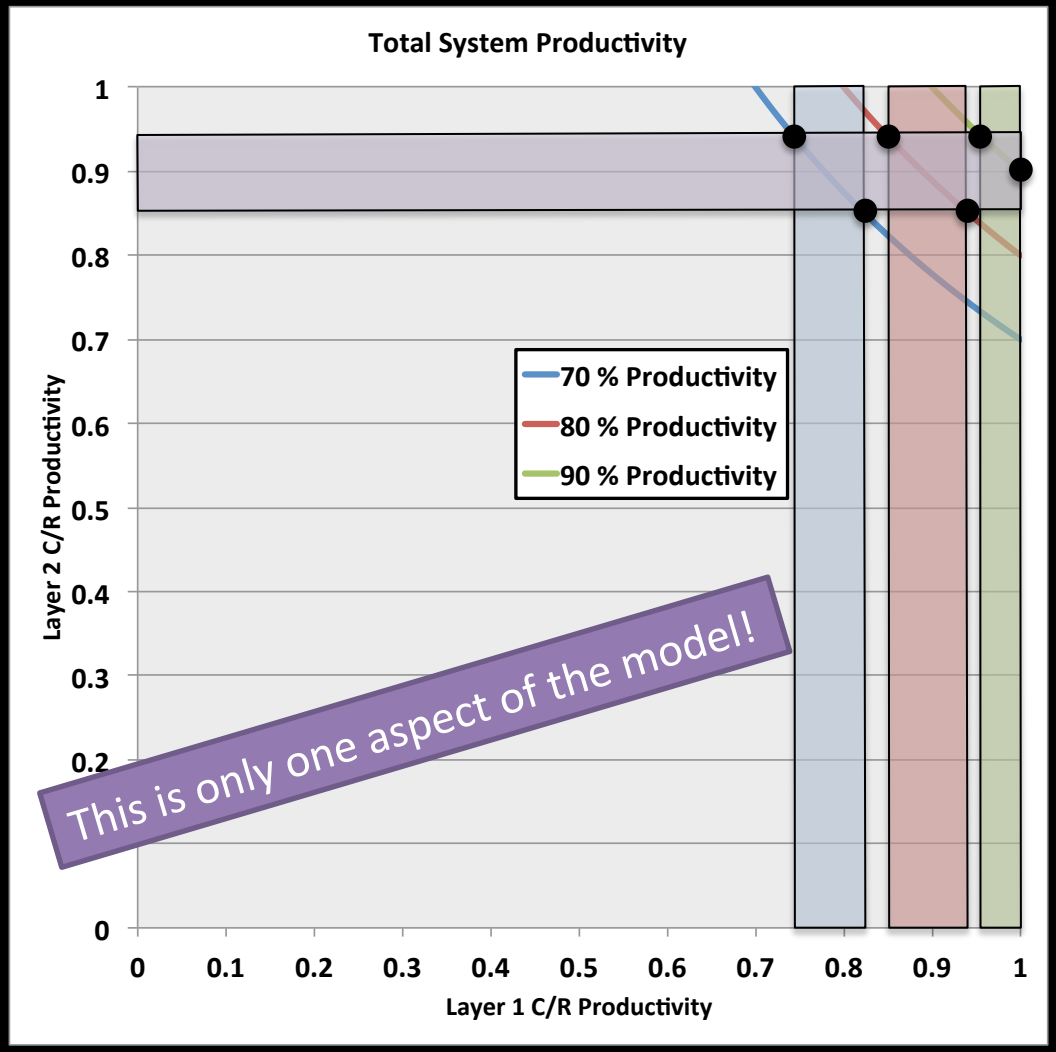
Multi-layer Checkpoint / Restart Modeling

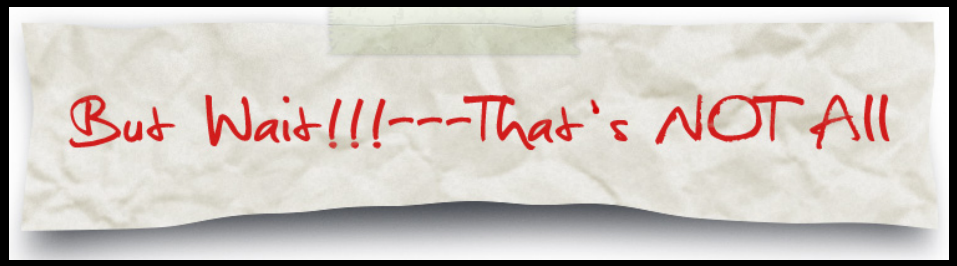
- If one of the layer's (Layer 2 in this example) is constrained . . .
  - By predicted / expected technology
  - Or other requirements (such as needed bandwidth, capacity, etc)



Multi-layer Checkpoint / Restart Modeling

- The model tells us how the other layer must perform to achieve desired system productivity levels

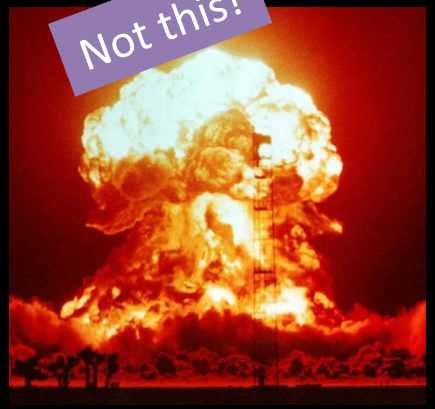




- Allows calculation of:
  - MTBF of each layer
  - Required checkpoint times
  - Required checkpoint restart times
  - Optimal checkpoint intervals <u>to each layer</u>
  - And more . . .
- This is a production-driven research exercise
  - We are doing this to make the applications run better

Conclusions and Thanks

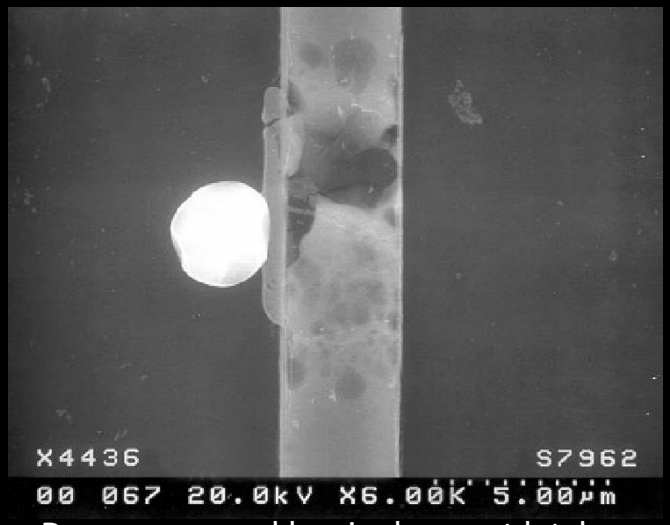
- Hopeful that hardware industry will help us out
  - We have to stay engaged to keep it on track
  - There are many opportunities for innovative research (big R is not dead)
- We need to test, test again, then test some more!
- Solutions will require an approach that looks at both SDC and hard fails
- Checkpointing alone is not sufficient
- Resilience is a cross-cutting field
  - Device physics, nuclear science, computer science, packaging, power, modeling and others
  - Requires the skills of lots of specialists, including the folks I work with that constantly assist me with my research
    - Sean Blanchard
    - Josip Loncaric
    - Heather Quinn
    - Andy DuBois
    - Dave DuBois
    - Sarah Michalak
- Thank you for your time and thank you for having me



Backup slides . . .

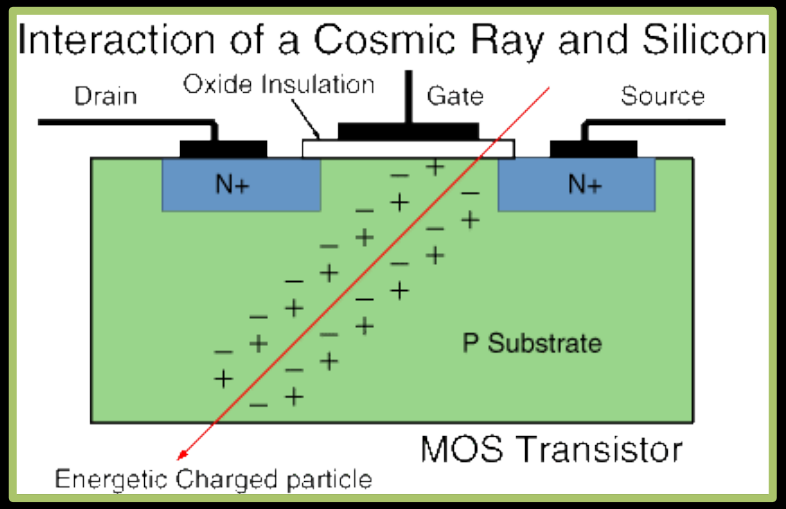
Memory Trends

- SEU per bit is decreasing
  - But . . . multi-bit upset probability is increasing
    - When particles strike, they shower a small area
    - This is why logical memory addresses are physically distributed in the chip
    - But as the area goes down, the number of bits in that shower area increases
- SEU per latch bit is decreasing
  - But . . . number of latches is growing



Damage caused by single event latchup

Becker, H.N.; Miyahira, T.F.; Johnston, A.H.; , "Latent damage in CMOS devices from single-event latchup," Nuclear Science, IEEE Transactions on , vol.49, no.6, pp. 3009- 3015, Dec 2002



Aerospace Corporation

We aren't dealing with latchup, but I am not aware of a picture like this for memory stuck bits

SEC/DED is good, DEC/TED must be great!

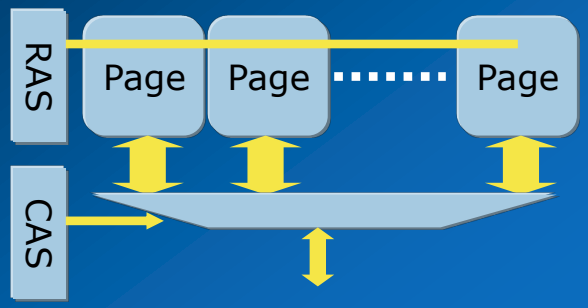
- Quinn reports "burst errors" in DIMMs where "5-all" of the bits in a page were changed
  - Attributed to strikes in the DRAM control circuitry
  - SEC/DEC won't catch this, nor will DEC/TED, etc.
  - Maybe we need Reed-Solomon error correction for the control circuitry. Unlikely to get it.
- DEC/TED likely to be wasted
- Need a way to reduce the susceptibility of the control circuitry

Quinn, H.; Graham, P.; Fairbanks, T.; , "SEEs Induced by High-Energy Protons and Neutrons in SDRAM," Radiation Effects Data Workshop (REDW), 2011 IEEE , vol., no., pp. 1-5, 25-29 July 2011

Los Alamos Neutron Science Center (LANSCE)

Revise DRAM Architecture

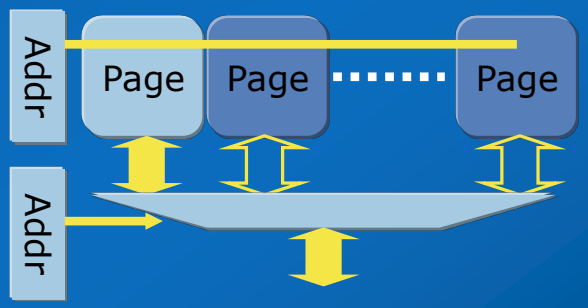
Traditional DRAM



Activates many pages
Lots of reads and writes (refresh)
Small amount of read data is used
Requires small number of pins

150 pJ/bit
(including Memory Controller)

New DRAM architecture



Activates few pages
Read and write (refresh) what is needed
All read data is used
Requires large number of IO's (3D)

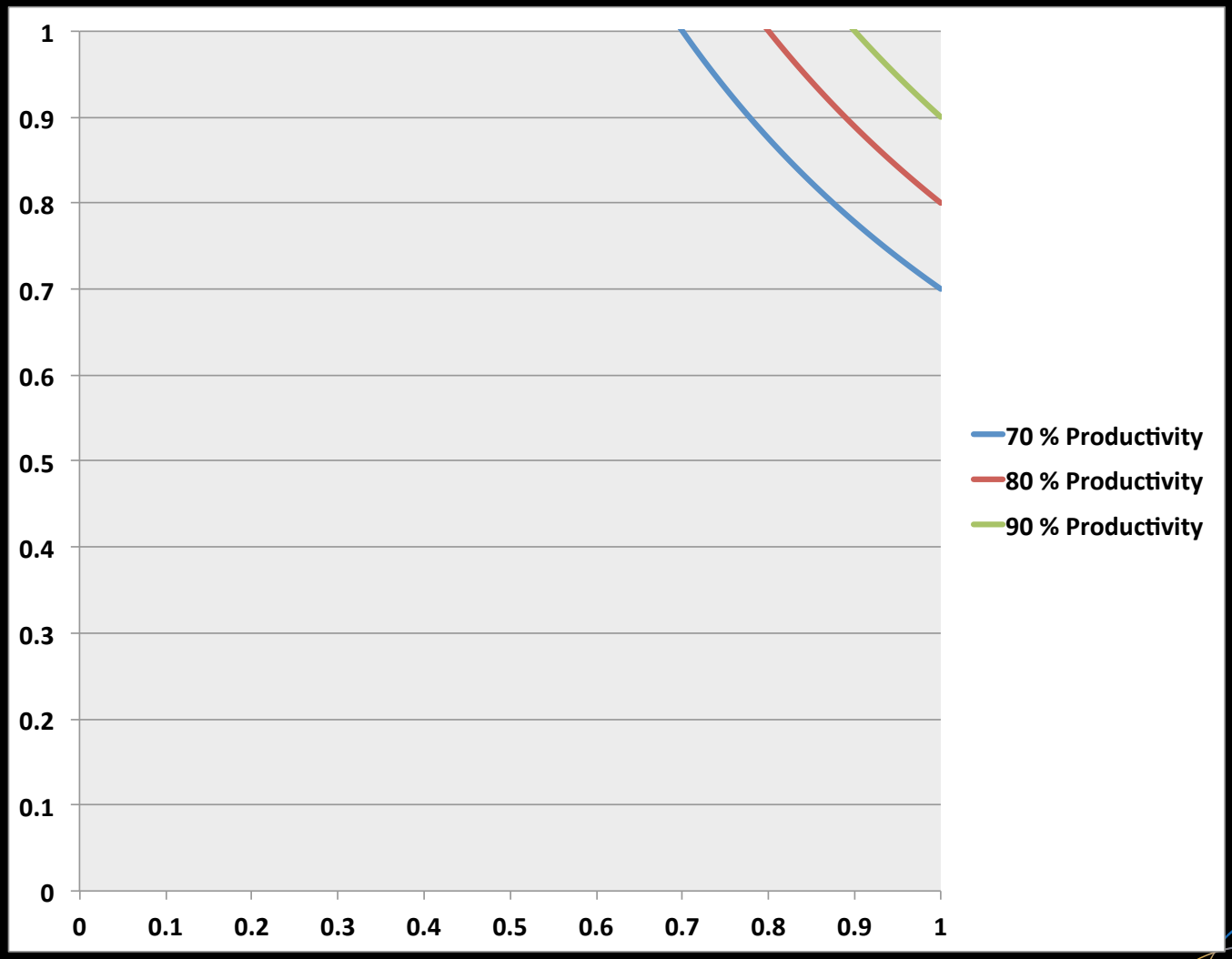
8-10 pJ/bit

Need to bring it down to 2 pJ/bit

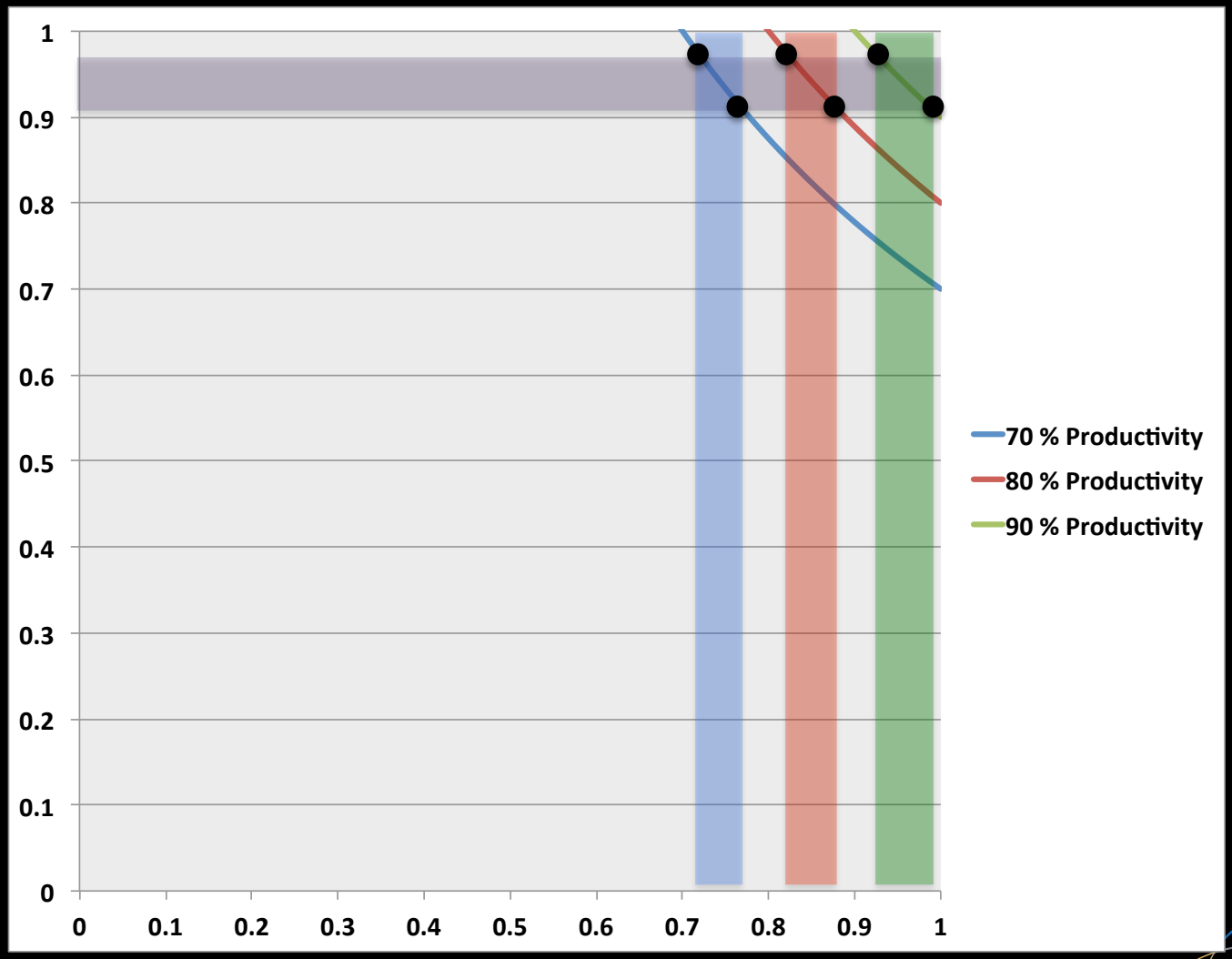
Multi-layer C/R

- Daly's models built upon Young's models
- Further improvements on Daly's models have been done to include checkpoint to medium A or medium B
- Our work is novel as it introduces the notion of asynchronous progress in the layers
- 1st Level C/R
  - P1 = useful but unreliable process
  - D1 = defensive overhead to improve reliability
  - X = P1 / (P1 + D1) = 1st level C/R productivity
- 2nd Level C/R may be needed
  - P2 = P1 + D1 useful but unreliable 2nd level process
  - D2 = defensive 2nd level overhead
  - Y = P2 / (P2 + D2) = 2nd level C/R productivity
- Overall productivity is X * Y

System productive utilization targets and how each layer's individual productive usage accounts for the whole



Assuming a constraint on one layer (due to capacity, bandwidth, expected technology, etc) we see how the other layer must perform



Provided Questions

- What have we learned about system fault tolerance, automatic hardware replacement, and data correction?
  - Fail in place is the way to go
- How can applications and algorithms be more resilient to undetected errors?
  - I'm not entirely sure yet, but I am sure it starts with understanding where an application is vulnerable.
- What have we learned about resiliency and process concurrency that will help us reduce data movement and disk I/O?
  - Resilience is at its core redundancy. Perhaps we can intelligently target redundancy. There certainly is an interrelationship between performance, power, and resilience. We need a model for this!
- What is the impact of non-determinism on debugging and quality assurance?
  - Potentially large, depending on how large the non-determinism is. Do we need to move to "run it twice" for algorithm development phases? Unit testing?
- Code creation: can we prevent bugs/race conditions/deadlocks rather than attempt to debug them after they occur?
  - I have nothing intelligent to say about this.
- There appears to be a community acceptance that we need a resilience API. What is the best way to get started?
  - Programmers, pizzas, sodas, closed doors, no meetings, couple months.
  - Really this isn't hard. It's been done at least once already. Problem is and probably always will be a community adoption problem.
  - Do we need to start putting into our RFPs that a vendor's system components will play nicely with this Resilience API?