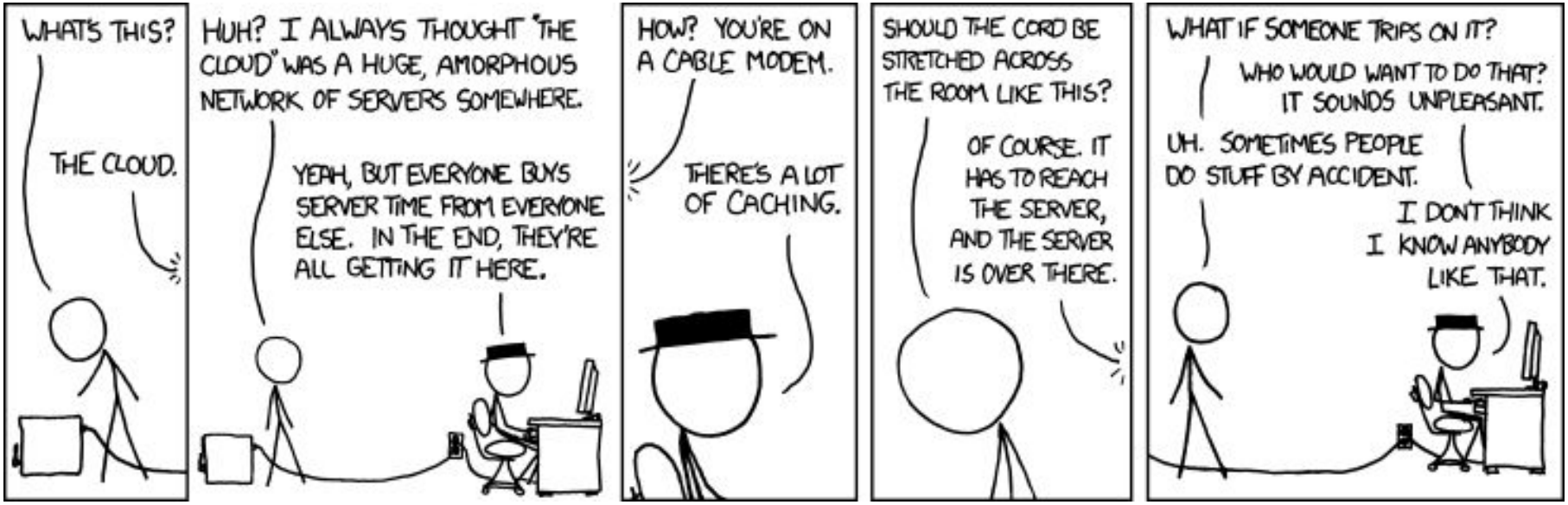
Caches IV

CSE 351 Summer 2021

Instructor:
Mara Kirdani-Ryan

Teaching Assistants:
Kashish Aggarwal
Nick Durand
Colton Jobs
Tim Mandzyuk



http://xkcd.com/908/

Gentle, Loving Reminders

- Unit Summary #2 Due tonight!
  - Floorplan & Design Doc in Task #1
  - Reflection in Task #2
  - Question responses in Task #3
- hw15 tonight, hw16 wednesday, hw17 friday!
  - Reach out if this isn't reasonable
- Lab 4 due next Monday (8/9)
  - All about caches!

Learning Objectives

Understanding this lecture means you can:

- Differentiate between different cache write policies
  - Write-back, Write-Through, Write-Allocate
- Optimize an algorithm for the memory hierarchy
  - Lab 4!
- Explain *Positivism* to someone outside of academia & computing, along with some critiques

We've mostly focused on cache reads...

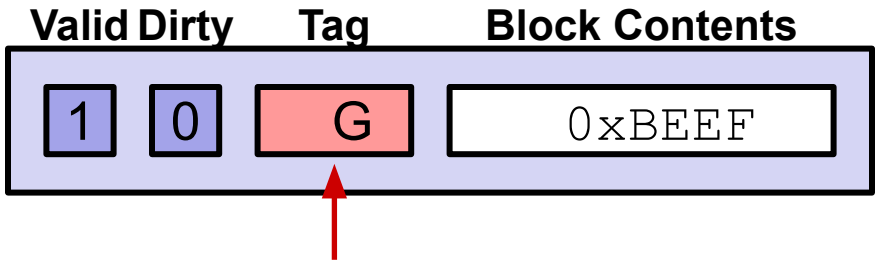
What about writes?

- Multiple copies of data may exist (caches, memory)
- What to do on a write-hit?
  - Write-through: write immediately to next level
  - Write-back: defer write to next level until line is replaced
    - Must track which cache lines have been modified ("*dirty bit*")
- What to do on a write-miss?
  - Write allocate: ("fetch on write") load into cache, then execute the write-hit policy
    - Good if more writes or reads to the location follow
  - No-write allocate: ("write around") just write immediately to next level
- Typical caches:
  - Write-back + Write allocate, usually
  - Write-through + No-write allocate, occasionally

Write-back, Write Allocate Example

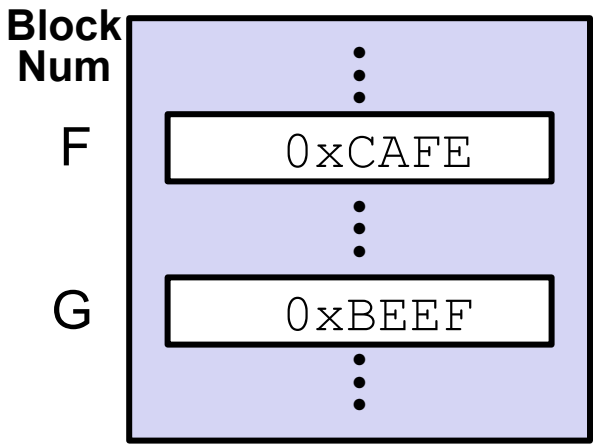
Note: While unrealistic, this example assumes that all requests have offset 0 and are for a block's worth of data.

Cache:



There is only one set in this tiny cache, so the tag is the entire block number!

Memory:

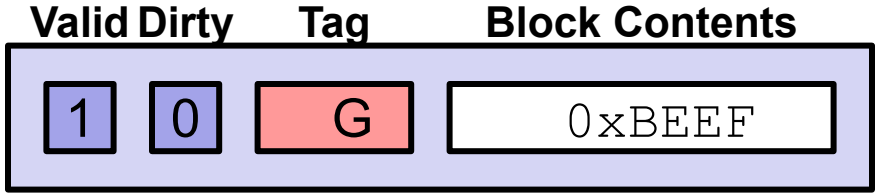


Write-back, Write Allocate Example

Not valid x86, just using block num instead of full byte address to keep the example simple

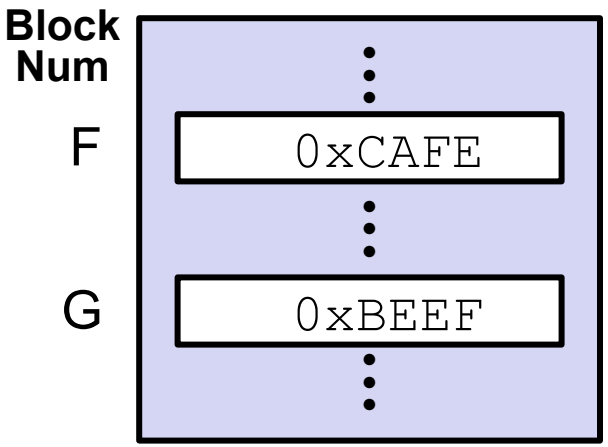
1) `mov $0xFACE,`
(F) Write Miss!

Cache:



Step 1: Bring **F** into cache

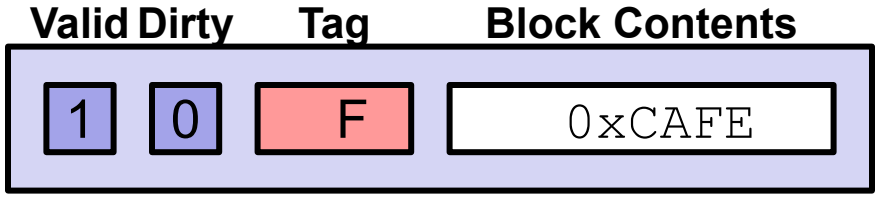
Memory:



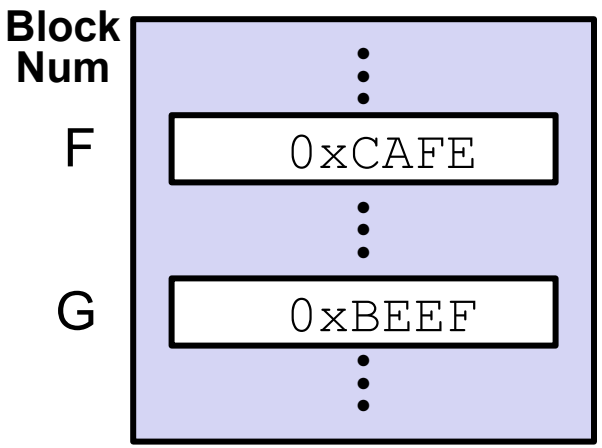
Write-back, Write Allocate Example

1) `mov $0xFACE,`
(F) Write Miss

Cache:



Memory:



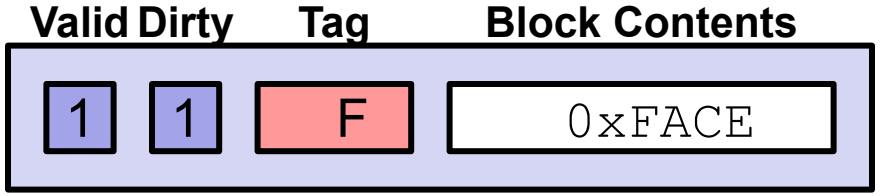
Step 1: Bring **F** into cache

Step 2: Write `0xFACE` to cache only and set the dirty bit

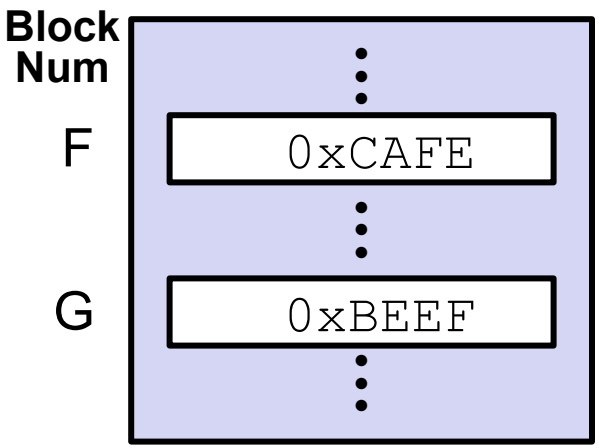
Write-back, Write Allocate Example

1) `mov $0xFACE,`
(F) Write Miss

Cache:



Memory:



Step 1: Bring **F** into cache

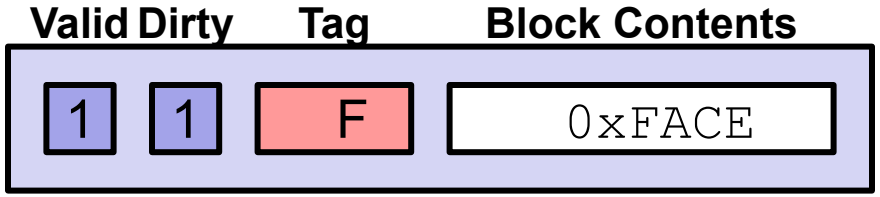
Step 2: Write `0xFACE` to cache only and set the dirty bit

Write-back, Write Allocate Example

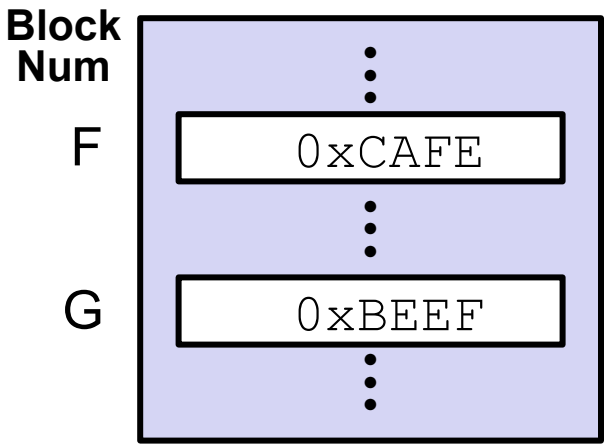
1) `mov $0xFACE,` (F) Write Miss

2) `mov $0xFEED,` (F) Write Hit!

Cache:



Memory:



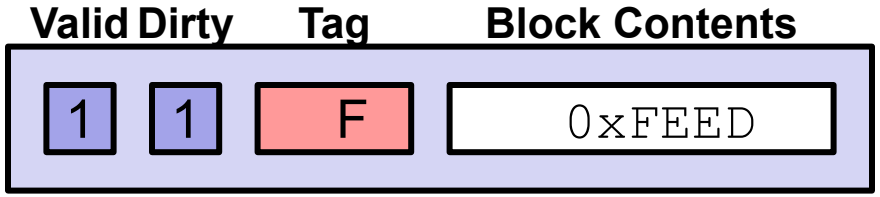
Step: Write `0xFEED` to cache only (and set the dirty bit)

Write-back, Write Allocate Example

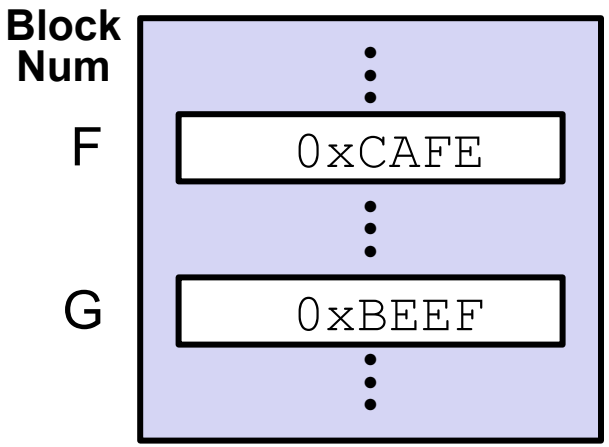
1) `mov $0xFACE,` (F) Write Miss

2) `mov $0xFEED,` (F) Write Hit

Cache:



Memory:



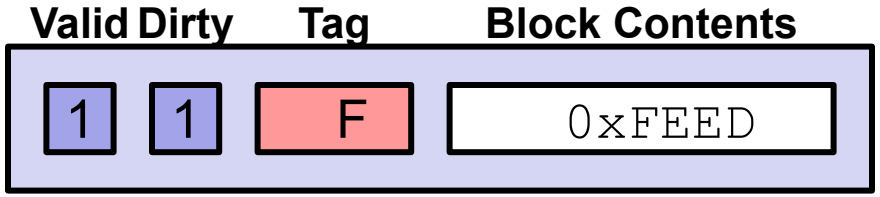
Write-back, Write Allocate Example

1) `mov $0xFACE,` (F) Write Miss

2) `mov $0xFEED,` (F) Write Hit

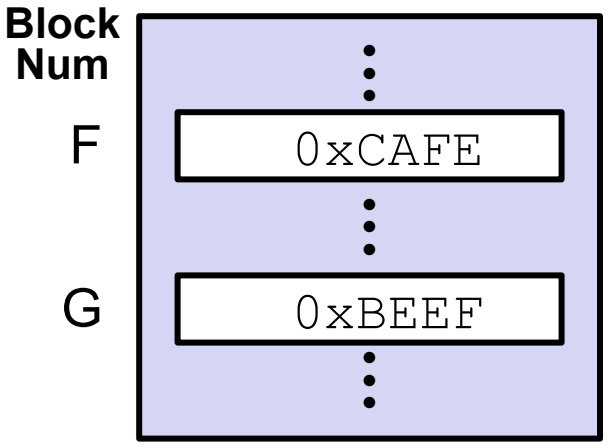
3) `mov (G), %ax` Read Miss!

Cache:



Step 1: Write **F** back to memory since it is dirty

Memory:



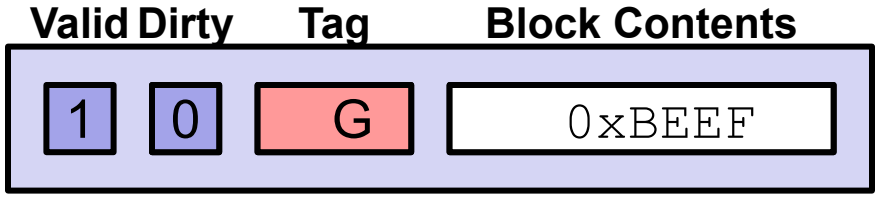
Write-back, Write Allocate Example

1) `mov $0xFACE,` (F) Write Miss

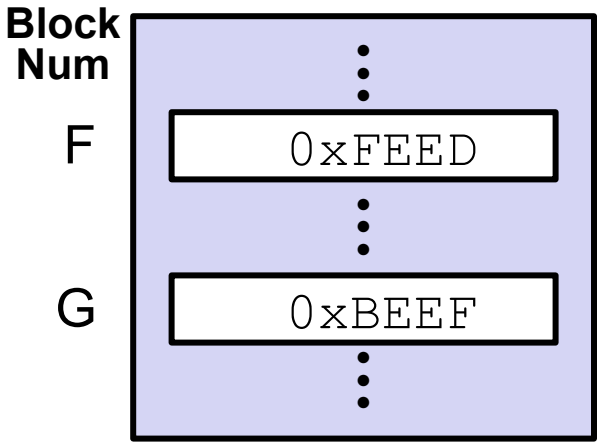
2) `mov $0xFEED,` (F) Write Hit

3) `mov (G), %ax` Read Miss

Cache:



Memory:



Step 1: Write **F** back to memory since it is dirty

Step 2: Bring **G** into the cache so that we can copy it into `%ax`

Checking in:
Write-back/Write-Allocate

Cache Simulator

- Want to play around with cache parameters and policies? Check out our cache simulator!
  - https://courses.cs.washington.edu/courses/cse351/caches im/

- Way to use:
  - Take advantage of "explain mode" and navigable history to test your own hypotheses and answer your own questions
  - Self-guided Cache Sim Demo posted along with Section 6
  - Will be used in hw16 – Lab 4 Preparation

Checking in! [Cache IV]

- Which cache statements is FALSE?

🐶 **We can reduce compulsory misses by decreasing our block size**

🐈 **We can reduce conflict misses by increasing associativity**

🐑 **A write-back cache will save time for code with good temporal locality on writes**

🦄 **A write-through cache will always match data with the memory hierarchy level below it**

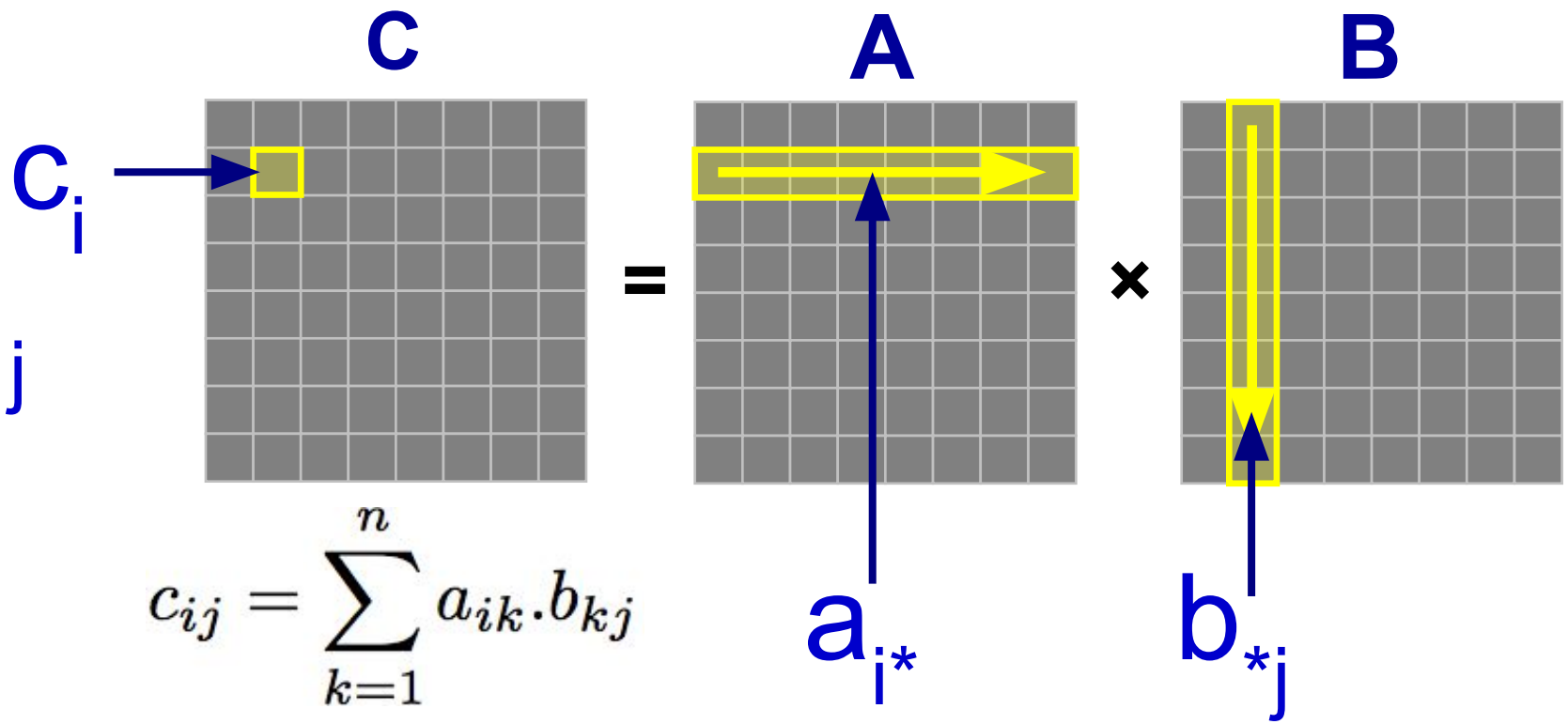
🥶 **Help!**

Optimizations for the Memory Hierarchy

- Memory Hierarchy assumes code with locality
  - Spatial: access data contiguously
  - Temporal: make sure access to the same data is not too far apart in time

- How can you achieve locality?
  - Adjust memory accesses in *code* (software) to improve miss rate (MR)
    - Requires knowledge of *both* how caches work as well as your system's parameters
  - Proper choice of algorithm
  - Loop transformations

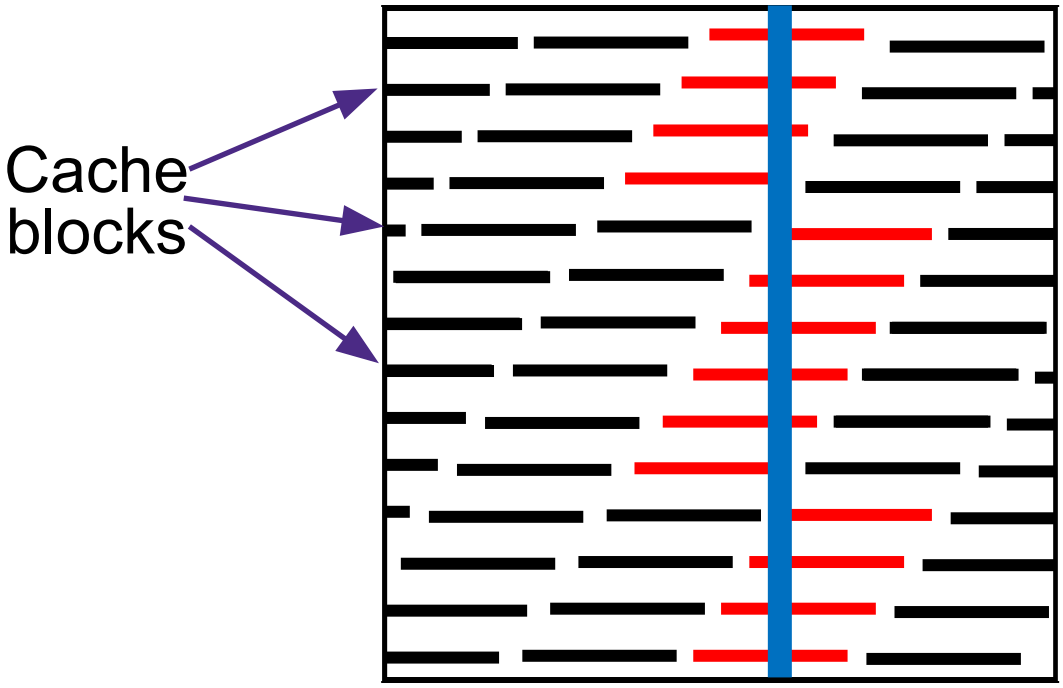
Example: Matrix Multiplication



$$c_{ij} = \sum_{k=1}^{n} a_{ik}.b_{kj}$$

Matrices in Memory

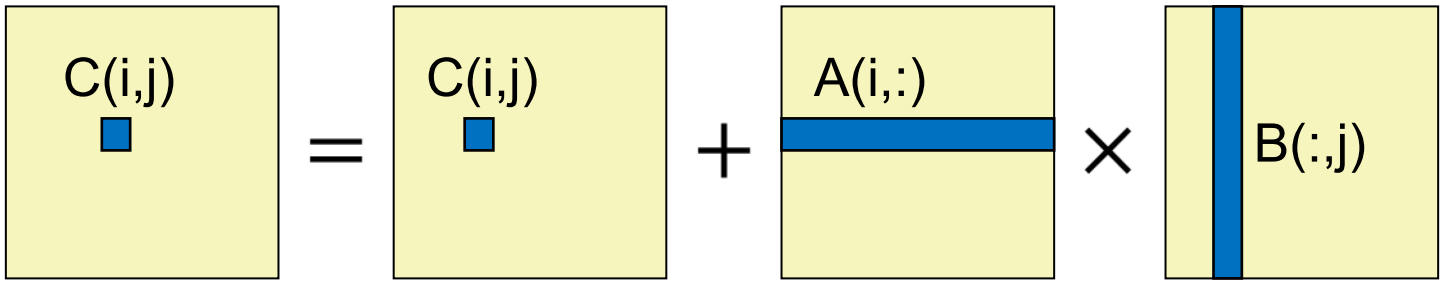
- How do cache blocks fit into this scheme?
  - Row major matrix in memory:



COLUMN of matrix (blue) is spread among cache blocks shown in red

Naïve Matrix Multiply

```
# move along rows of A
for (i = 0; i < n; i++)
  # move along columns of B
  for (j = 0; j < n; j++)
    # EACH k loop reads row of A, col of B
    # Also read & write c(i,j) n times
    for (k = 0; k < n; k++)
      c[i*n+j] += a[i*n+k] * b[k*n+j];
```

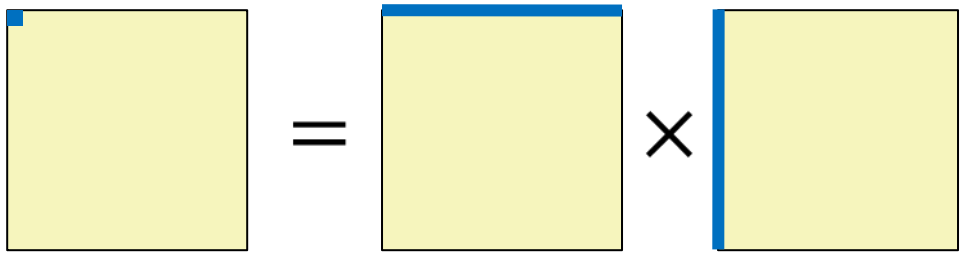


Cache Miss Analysis (Naïve)

Ignoring matrix `c`

- Scenario Parameters:
  - Square matrix ($n \times n$), elements are `doubles`
  - Cache block size K = 64 B = 8 `doubles`
  - Cache size $C \ll n$ (much smaller than n)

- Each iteration:
  - $\frac{n}{8} + n = \frac{9n}{8}$ misses

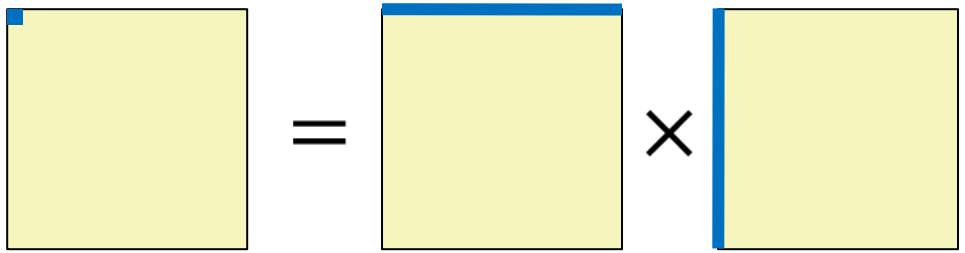


Cache Miss Analysis (Naïve)

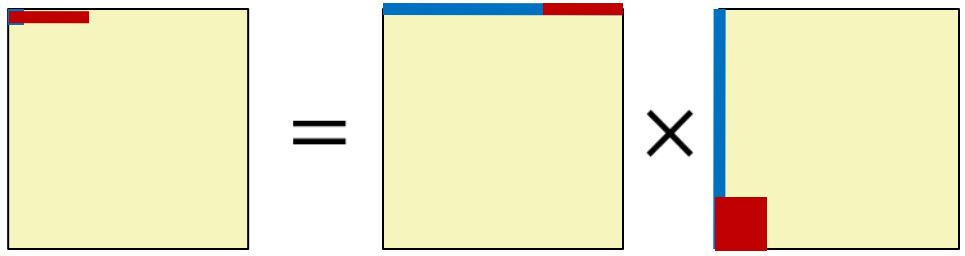
Ignoring matrix `C`

- Scenario Parameters:
  - Square matrix ($n \times n$), elements are `doubles`
  - Cache block size K = 64 B = 8 `doubles`
  - Cache size $C \ll n$ (much smaller than n)

- Each iteration:
  - $\frac{n}{8} + n = \frac{9n}{8}$ misses



  - Afterwards in cache: (schematic)

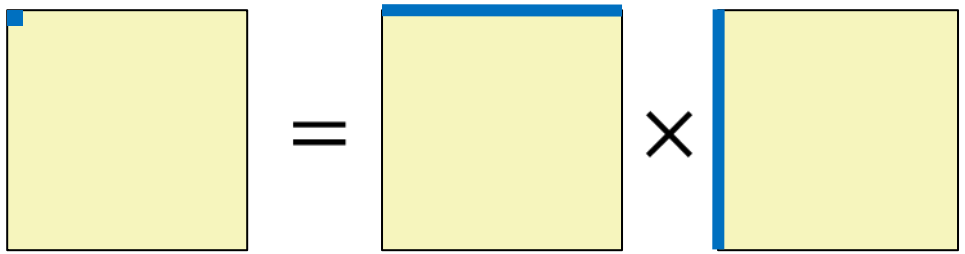


Cache Miss Analysis (Naïve)

Ignoring matrix `c`

- Scenario Parameters:
  - Square matrix ($n \times n$), elements are `doubles`
  - Cache block size K = 64 B = 8 `doubles`
  - Cache size $C \ll n$ (much smaller than n)

- Each iteration:
  - $\frac{n}{8} + n = \frac{9n}{8}$ misses



- Total misses: $\frac{9n}{8} \times n^2 = \frac{9}{8}n^3$

once per product matrix element

Linear Algebra to the Rescue (1)

This is extra (non-testable) material

- Can get the same result of a matrix multiplication by splitting the matrices into smaller submatrices (matrix "blocks")
- For example, multiply two 4×4 matrices:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \text{ with } B \text{ defined similarly.}$$

$$AB = \begin{bmatrix} (A_{11}B_{11} + A_{12}B_{21}) & (A_{11}B_{12} + A_{12}B_{22}) \\ (A_{21}B_{11} + A_{22}B_{21}) & (A_{21}B_{12} + A_{22}B_{22}) \end{bmatrix}$$

Linear Algebra to the Rescue (2)

This is extra (non-testable) material

C_{11}	C_{12}	C_{13}	C_{14}
C_{21}	C_{22}	C_{23}	C_{24}
C_{31}	C_{32}	C_{43}	C_{34}
C_{41}	C_{42}	C_{43}	C_{44}

A_{11}	A_{12}	A_{13}	A_{14}
A_{21}	A_{22}	A_{23}	A_{24}
A_{31}	A_{32}	A_{33}	A_{34}
A_{41}	A_{42}	A_{43}	A_{14}

4

B_{11}	B_{12}	B_{13}	B_{14}
B_{21}	B_{22}	B_{23}	B_{24}
B_{32}	B_{32}	B_{33}	B_{34}
B_{41}	B_{42}	B_{43}	B_{44}

Matrices of size $n \times n$, split into 4 blocks of size r (n=4r)

$$C_{22} = A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32} + A_{24}B_{42} = \sum_k A_{2k} * B_{k2}$$

- Multiplication operates on small "block" matrices
  - Choose size so that they fit in the cache!
  - This technique called "*cache blocking*"

Blocked Matrix Multiply

- Blocked version of the naïve algorithm:

```
# move by rxr BLOCKS now
for (i = 0; i < n; i += r)
  for (j = 0; j < n; j += r)
    for (k = 0; k < n; k += r)
      # block matrix multiplication
      for (ib = i; ib < i+r; ib++)
        for (jb = j; jb < j+r; jb++)
          for (kb = k; kb < k+r; kb++)
            c[ib*n+jb] += a[ib*n+kb]*b[kb*n+jb];
```

  - r = block matrix size (assume r divides n evenly)

Cache Miss Analysis (Blocked)

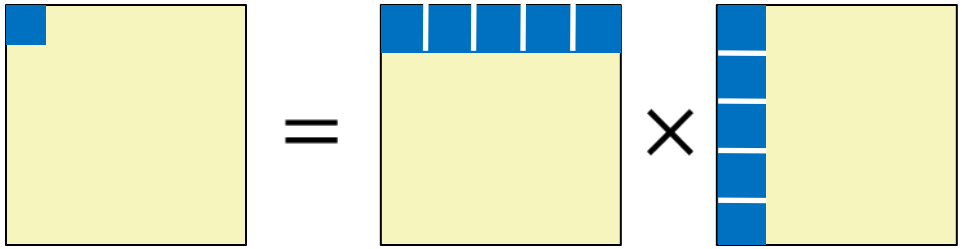
Ignoring matrix C

- Scenario Parameters:
  - Cache block size K = 64 B = 8 doubles
  - Cache size $C \ll n$ (much smaller than n)
  - Three blocks ■ $(r \times r)$ fit into cache: $3r^2 < C$

r^2 elements per block, 8 per cache block

n/r blocks

- Each block iteration:
  - $r^2/8$ misses per block
  - $2n/r \times r^2/8 = nr/4$



n/r blocks in row and column

Cache Miss Analysis (Blocked)

Ignoring matrix C

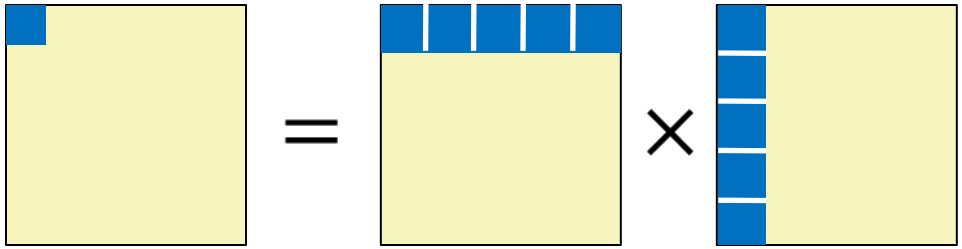
- Scenario Parameters:
  - Cache block size K = 64 B = 8 doubles
  - Cache size $C \ll n$ (much smaller than n)
  - Three blocks ■ $(r \times r)$ fit into cache: $3r^2 < C$

r^2 elements per block, 8 per cache block

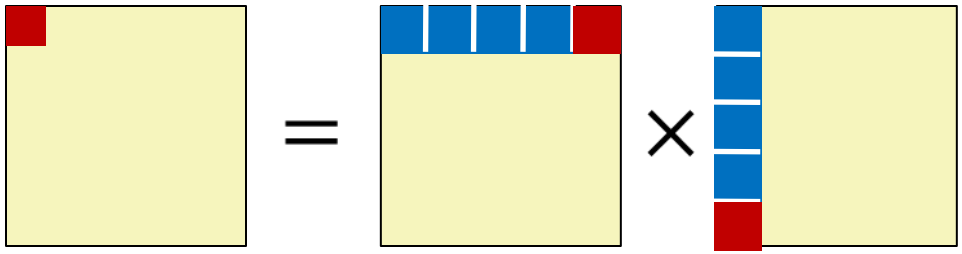
n/r blocks

- Each block iteration:
  - $r^2/8$ misses per block
  - $2n/r \times r^2/8 = nr/4$

n/r blocks in row and column



- Afterwards in cache (schematic)



Cache Miss Analysis (Blocked)

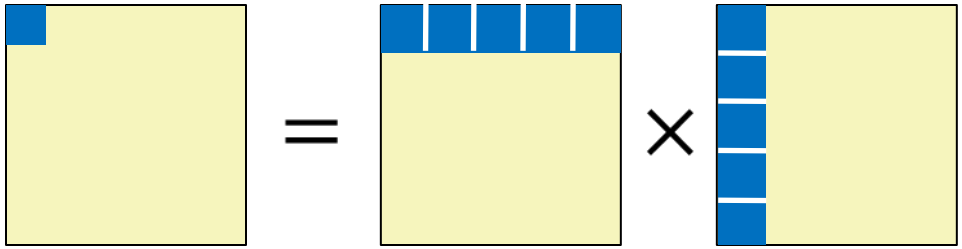
Ignoring matrix C

Scenario Parameters:

- Cache block size K = 64 B = 8 doubles
- Cache size C ≪ n (much smaller than n)
- Three blocks ■ (r×r) fit into cache: $3r^2<C$

r^2 elements per block, 8 per cache block

n/r blocks



Each block iteration:

- $r^2/8$ misses per block
- $2n/r \times r^2/8 = nr/4$

n/r blocks in row and column

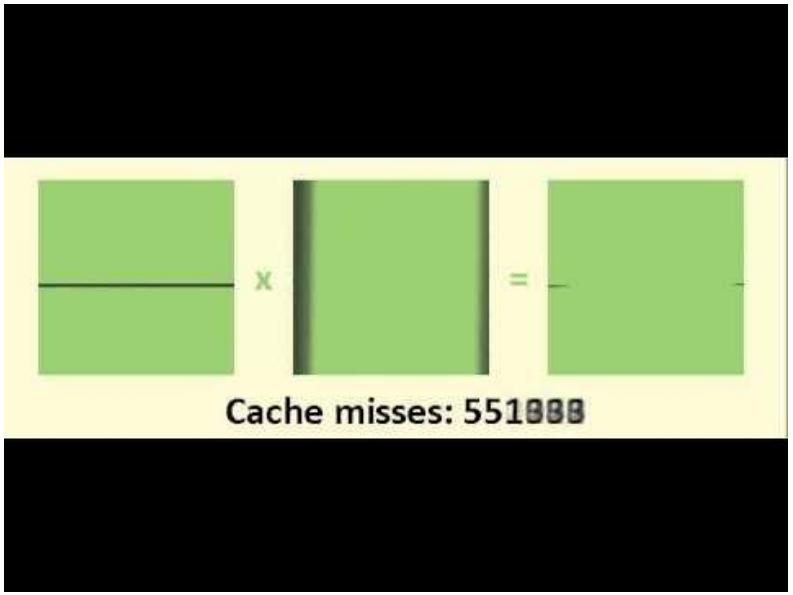
Total misses:

- $nr/4 \times (n^2/r^2) = n^3/(4r)$

Matrix Multiply Visualization

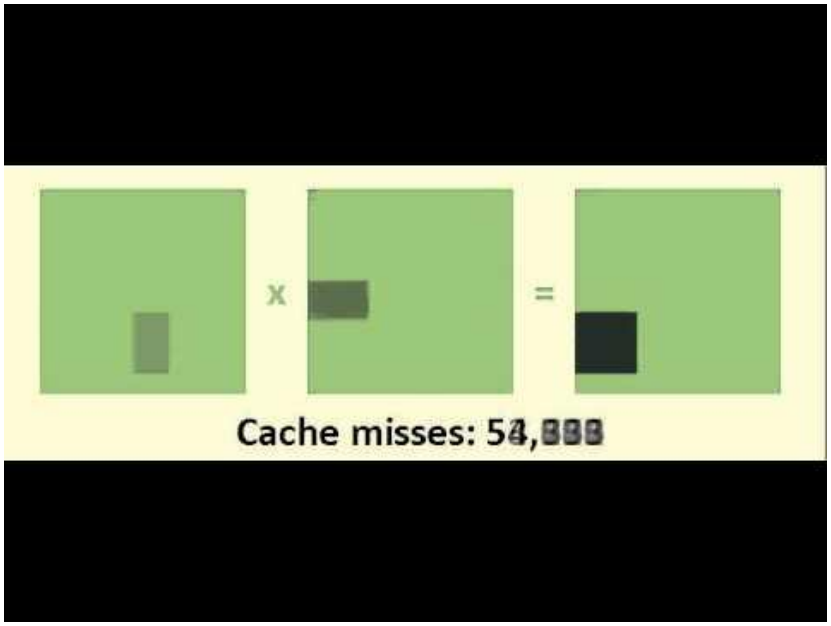
- Here $n = 100$, $C = 32$ KiB, $r = 30$

Naïve:



≈ 1,020,000
cache misses

Blocked:



≈ 90,000
cache misses

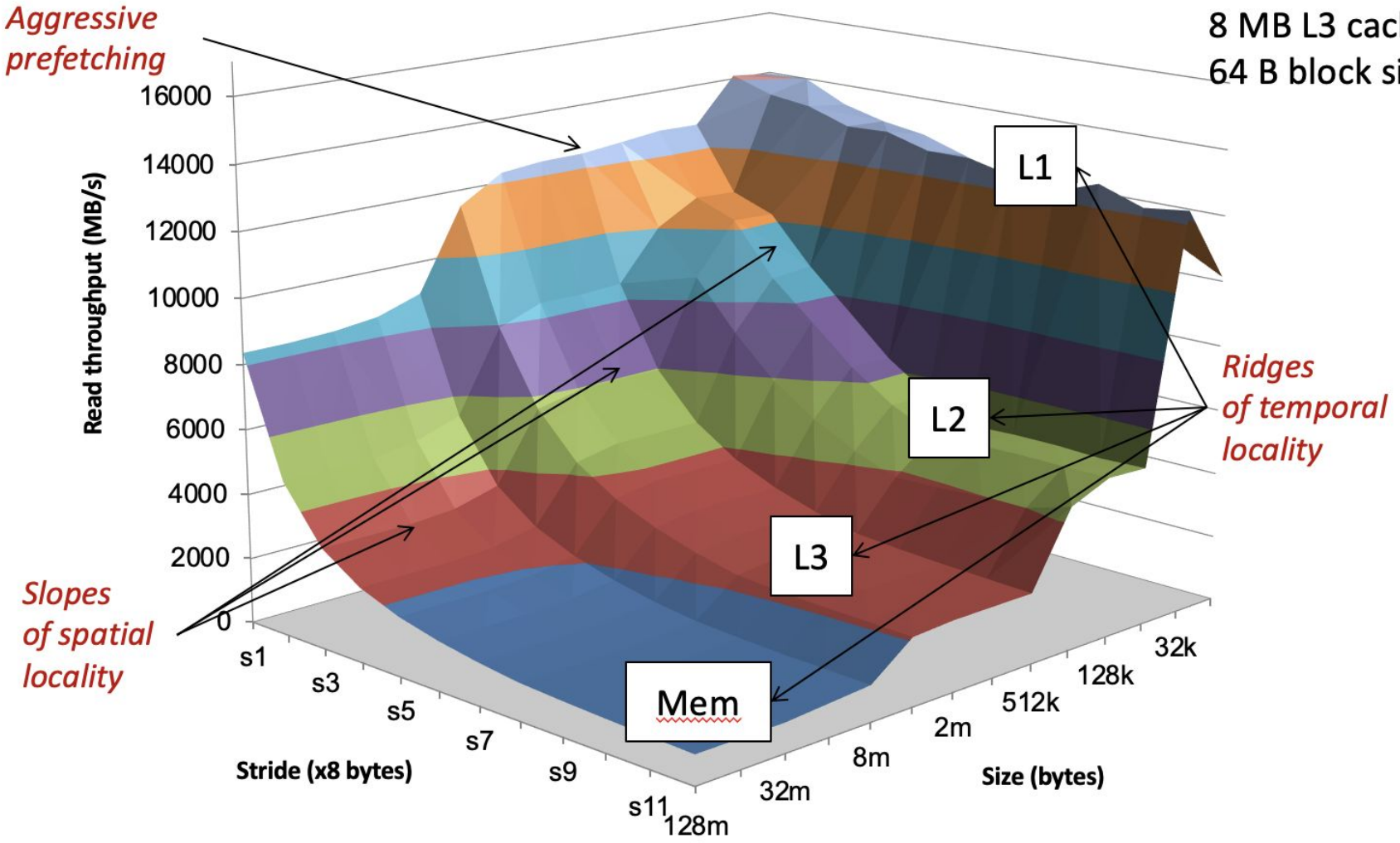
Checking in:
Matrix-Multiply

Cache-Friendly Code

- Programmer can optimize for cache performance
  - How data structures are organized
  - How data are accessed
    - Nested loop structure
    - Blocking is a general technique
- All systems favor "cache-friendly code"
  - Getting *absolute optimum* performance is very platform specific
    - Cache size, cache block size, associativity, etc.
  - Can get most of the advantage with generic code
    - Keep working set reasonably small (temporal locality)
    - Use small strides (spatial locality)
    - Focus on inner loop code

The Memory Mountain

Core i7 Haswell
2.1 GHz
32 KB L1 d-cache
256 KB L2 cache
8 MB L3 cache
64 B block size



Learning About Your Machine

- **Linux:**
  - `lscpu`
  - ls /sys/devices/system/cpu/cpu0/cache/index0/
    - Example: cat /sys/devices/system/cpu/cpu0/cache/index*/size
- **Windows:**
  - `wmic memcache get <query>` (all values in KB)
  - Example: `wmic memcache get MaxCacheSize`
- Modern processor specs: http://www.7-cpu.com/

Positivism

Last time(s):

- Broad tendency to derive choices about metric and measurement from ideology
  - Then, measurement choices define success and influence the structure that's created
  - Efficiency → Program Performance → Caches
  - Not always bad! Just ideological, and worth noting.

Last time(s):

- Broad tendency to derive choices about metric and measurement from ideology
  - Then, measurement choices define success and influence the structure that's created
  - Efficiency → Program Performance → Caches
  - Not always bad! Just ideological, and worth noting.
- CS's tends to position itself as segregated:
  - Objective, countering the world's subjectivity
  - Neutral, countering the world's conflicting values

Last time(s):

- Broad tendency to derive choices about metric and measurement from ideology
  - Then, measurement choices define success and influence the structure that's created
  - Efficiency → Program Performance → Caches
  - Not always bad! Just ideological, and worth noting.
- CS's tends to position itself as segregated:
  - Objective, countering the world's subjectivity
  - Neutral, countering the world's conflicting values
- This positionality leads to harm when CS, inevitably, interacts with the world
  - Beautiful complexity seen as a threat to performance
  - Me: An anomaly, and a threat

Epistemology

First, Epistemology

- **Defn:** Philosophy surrounding knowledge, especially knowledge creation & validation
  - *I've used this word a lot*
  - *What do we know? How do we know it? When can we justify our beliefs?*
- *Epistemologically Valid:* True, given our beliefs and understandings of knowledge
  - This (in part) is how research is judged!
  - "Is what you say true, given our epistemology?"

It gets messy quick

- Ok, but when can we say something is valid?
  - *It depends on our epistemology!*
  - *When have we done enough to say something is true, and to justify that statement?*

It gets messy quick

- Ok, but when can we say something is valid?
  - *It depends on our epistemology!*
  - *When have we done enough to say something is true, and to justify that statement?*

- We could have something basic:
  - "Mara decides all validity"
  - *"Bring me your claims; I will determine their validity!"*
- Despite being (in part) how traditional education works, this wouldn't be so fun!
  - What if I'm out of town? What if I'm sick?

There's lots of different options here, obviously.

If you want to create knowledge, it's really important to think about where you stand.

Disclaimer:
As usual, this looking at is the discipline of CS, not every person in it.

Positivism: Defined

- *In plain language:*
  If we can measure it, it exists!
- Separation from the "knower" and the "known" -- "real science" is objective, cold, calculating
  - *Maybe this sounds familiar*

Positivism: Defined

- *In plain language:*
  If we can measure it, it exists!
- Separation from the "knower" and the "known" -- "real science" is objective, cold, calculating
  - *Maybe this sounds familiar*
- Claim: Sociology is objective!
  - The "knower" is wholly separate from the "known"; the scientist can (and should) remove their "common sense" understandings from their research
  - Replication isn't an issue --- multiple sociologists approach from the same removed, objective space
  - **We can study people as objectively as we study particles**

An overview of positivism

	Positivism
The observer	Can and must be independent
Human Interests	Should be irrelevant
Explanations	Must demonstrate causality
Research progresses through	Hypothesis and Deductions
Concepts	Need to be operationalized, so that they can be measured
Units of Analysis...	Should be reduced to simplest terms
We can generalize through	Statistical Probability
Sampling requires	Large numbers, selected randomly

An overview of positivism

	Positivism
The observer	Can and must be independent
Human Interests	Should be irrelevant
Explanations	Must demonstrate causality
Research progresses through	Hypothesis and Deductions
Concepts	Need to be operationalized, so that they can be measured
Units of Analysis...	Should be reduced to simplest terms
We can generalize through	Statistical Probability
Sampling requires	Large numbers, selected randomly

What's implicit?

Assumptions of Positivism

1. Researchers can act in a way that's independent
   - *But, how removed can you really be?*
   - *Can anyone be wholly unbiased?*

Assumptions of Positivism

1. Researchers can act in a way that's independent
   - *But, how removed can you really be?*
   - *Can anyone be wholly unbiased?*
2. Research motivations have no effect on results
   - *But, someone's paying for it, right?*
   - *Can anyone separate from their motivations?*

Assumptions of Positivism

1. Researchers can act in a way that's independent
   - *But, how removed can you really be?*
   - *Can anyone be wholly unbiased?*
2. Research motivations have no effect on results
   - *But, someone's paying for it, right?*
   - *Can anyone separate from their motivations?*
3. We must be operationally defined to measure
   - *But, are the definitions objective?*
   - *Didn't we come up with them?*
   - *If we make a grouping, did we include everyone?*

Only one epistemology, and a problematic one!

Critical Theory

A “critical” theory may be distinguished from a “traditional” theory according to a specific practical purpose: a theory is critical to the extent that it seeks human “emancipation from slavery”, acts as a “liberating … influence”, and works “to create a world which satisfies the needs and powers of” human beings (Horkheimer 1972b)

Critical Theory, very briefly

- Seeks liberation for all people!
  - Break the chains of oppression for all people!

- **Positivism**: We should explain society
- **Critical Theory:** We should change society!
  - *Critically*, we should research society itself, situated in the histories of its creation
  - We should point out oppression with research!
  - *Activism:* we should also liberate with research

Critiques of Positivism

- Researchers bring their own biases to their work
  - *Designers, researchers, engineers bring ideology*
  - We need to be *reflexive*! Positivism isn't!
  - We should examine ourselves before we create!

Critiques of Positivism

- Researchers bring their own biases to their work
  - *Designers, researchers, engineers bring ideology*
  - We need to be *reflexive*! Positivism isn't!
  - We should examine ourselves before we create!
- Motivations for research affect results!
  - *What we find depends on what we were looking for*
  - We should make our motivations clear!
  - We should be motivated towards liberation!

Critiques of Positivism

- Researchers bring their own biases to their work
  - *Designers, researchers, engineers bring ideology*
  - We need to be *reflexive*! Positivism isn't!
  - We should examine ourselves before we create!
- Motivations for research affect results!
  - *What we find depends on what we were looking for*
  - We should make our motivations clear!
  - We should be motivated towards liberation!
- We affect what we create! Not objective!
  - *How we name what we build depends on us!*
  - We should build in an inclusive way!
  - We should involve who we're naming in the process

One Alternative: Interpretivism

- Reality and knowledge are socially constructed
  - *In part, by all researchers, designers, engineers*
  - *Findings are "interpreted", rather than "discovered"*
- No objective reality!
  - We have our interpretations, nothing more
  - We can have different interpretations
  - We can look for affinity between interpretations

It's spooky, but it means that different stories, from different people, can both be legitimate.

Which means we can call out racism, even if we haven't experienced oppression.

Positivism and Interpretivism

	Positivism	**Interpretivism**
The observer	Can and must be independent	Is part of what is being observed
Human Interests	Should be irrelevant	Are the main drivers of science
Explanations	Must demonstrate causality	Aim to increase general understanding of situations
Research progresses through	Hypothesis and Deductions	Gather rich data from which ideas are induced
Concepts	Need to be operationalized, so that they can be measured	Should incorporate stakeholder perspectives
Units of Analysis...	Should be reduced to simplest terms	May include the complexity of “whole” situations
We can generalize through	Statistical Probability	Theoretical abstraction
Sampling requires	Large numbers, selected randomly	Small numbers of cases, chosen for specific reasons

You get to pick!

It's *your* epistemology, after all!

It's less about caches, and more about what the represent

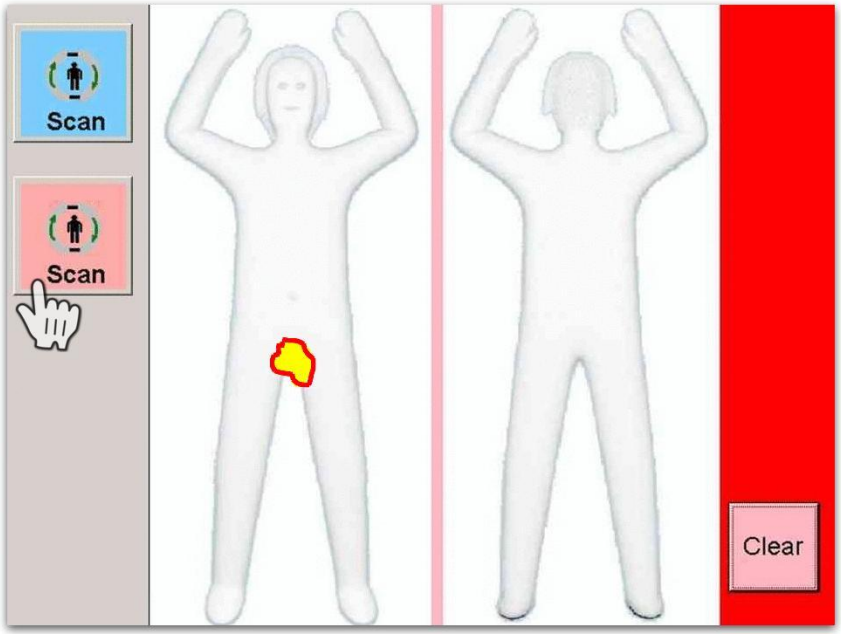
CS, Caches, Positivism

- We have an ideology (neoliberal capitalism)
- We pick a metric (program performance)
- We pick optimizations to succeed along that metric (caches)
- We remeasure, succeed along our metric, and claim objective performance improvements

- *This isn't too terrible, no human interfacing*

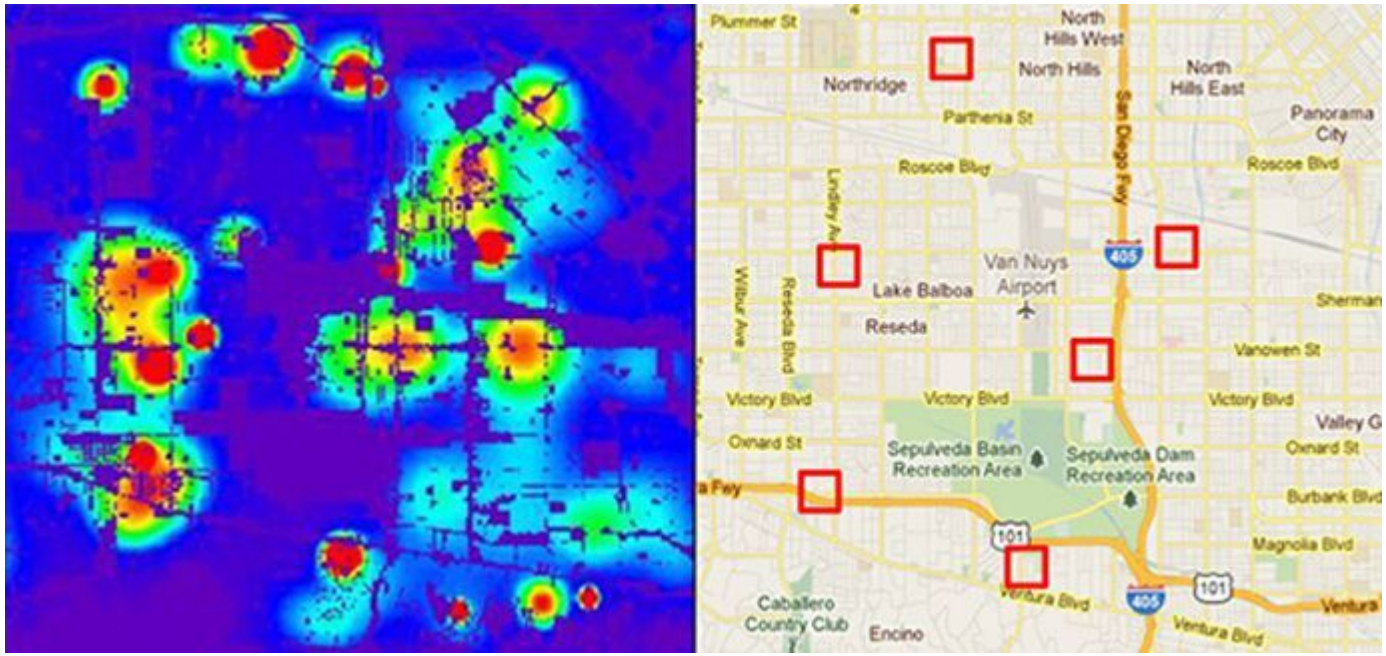
Positivism and Computing

- Need to operationalize gender to measure (binary)
- Unexamined researcher bias
- "Objective" scanners that claim high accuracy



Predictive Policing

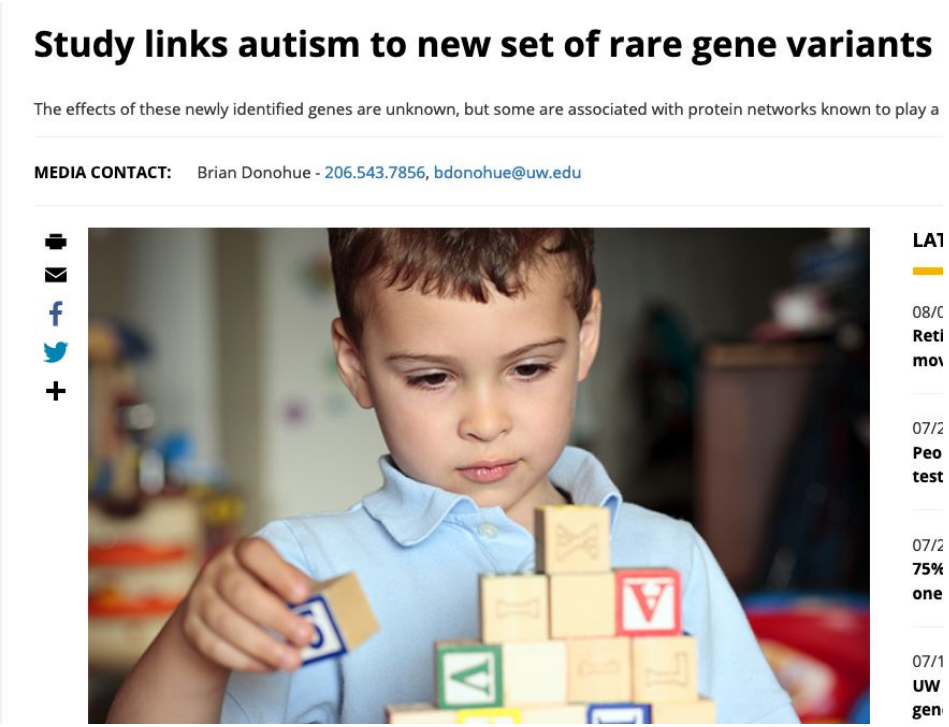
- "fair and accountable" algorithms that predict *where* crime might happen
  - Other algorithms predict who might be involved!
  - Claimed to be "fair, objective, accountable"



Assume that the researchers/engineers are free from bias

Isolating the ___ gene

- Massive 35k genome dataset
- "Our findings won't be brought into the clinic tomorrow"
  - *Nothing about us, without us*
- Positivist ideology, applied to a very, very messy, racist, sexist diagnosis



Positivism explicitly ignores anything systemic

The researcher is wholly objective, their motivations pure and unbiased

Please, be reflexive!

Understand, as much as you can, who you are before you build!

Don’t stop reflecting!

Ask for help! Talk with others!

Reflect as you build, before you release, after you release.

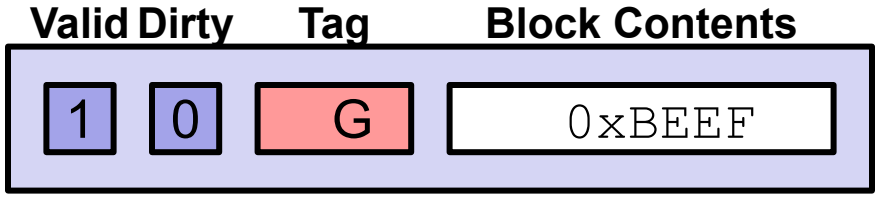
Write-back, Write Allocate Example

1) `mov 0xFACE,` F

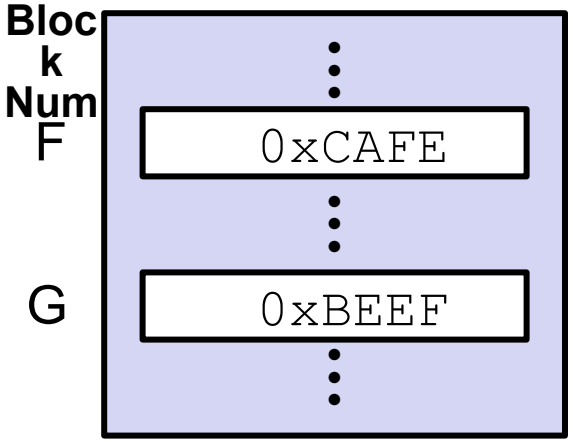
2) `mov 0xFEED,` F

3) `mov G, %ax`

Cache:



Memory:

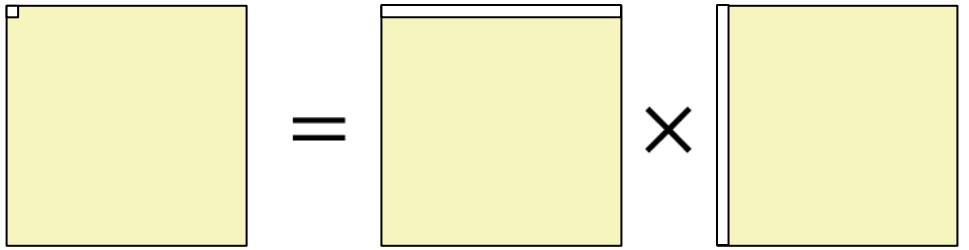


Cache Miss Analysis Comparison

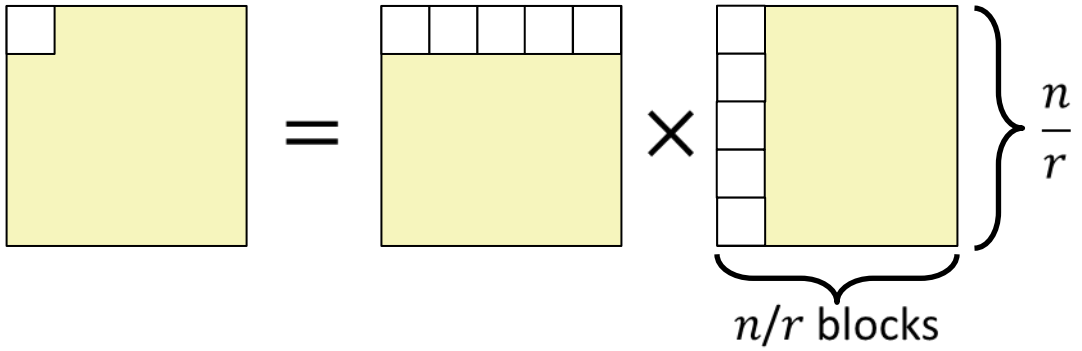
Ignoring matrix `C`

- Scenario Parameters:
  - Square matrix ($n \times n$) of `doubles`
  - Cache block size K = 64 B = 8 `doubles`
  - Cache size $C \ll n$ and three blocks ($r \times r$) fit into cache: $3r^2 < C$

- Naïve:

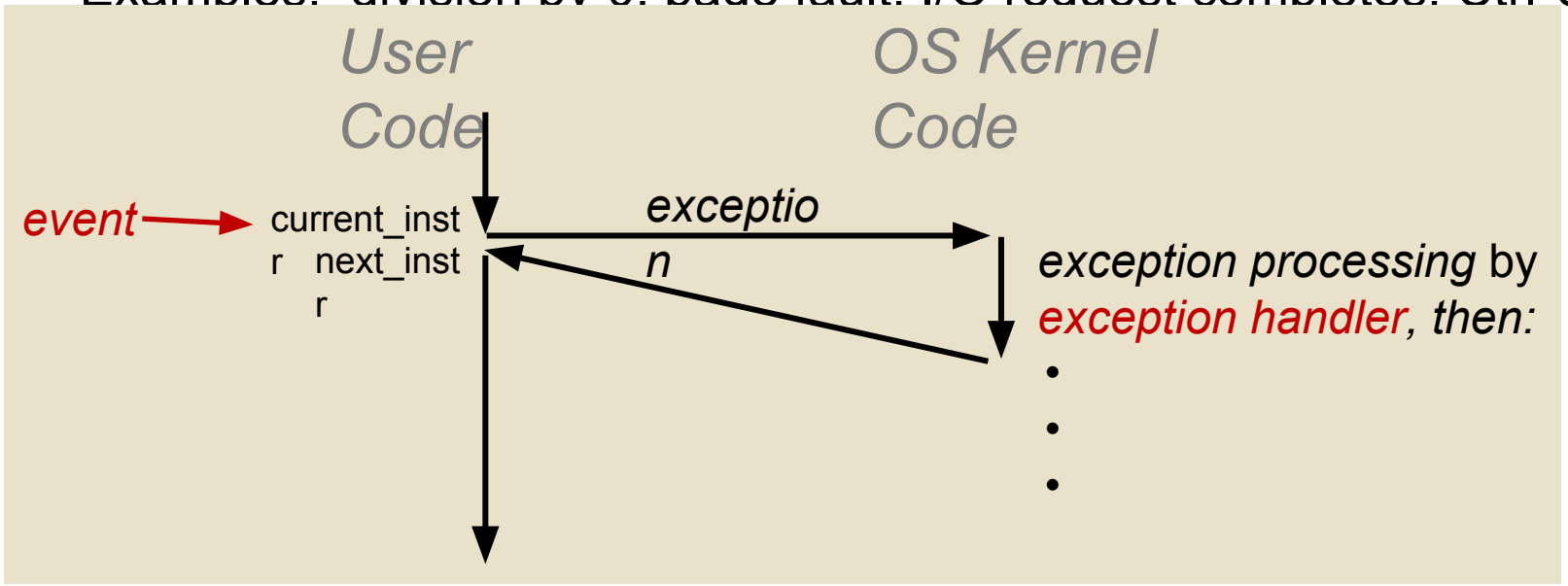


- Blocked:



Exceptions - Handout

- An *exception* is transfer of control to the operating system (OS) kernel in response to some *event* (*i.e.* change in processor state)
  - Kernel is the memory-resident part of the OS
  - Examples: division by 0, page fault, I/O request completes, Ctrl-C



- *How does the system know where to jump to in the OS?*

Notes Diagrams