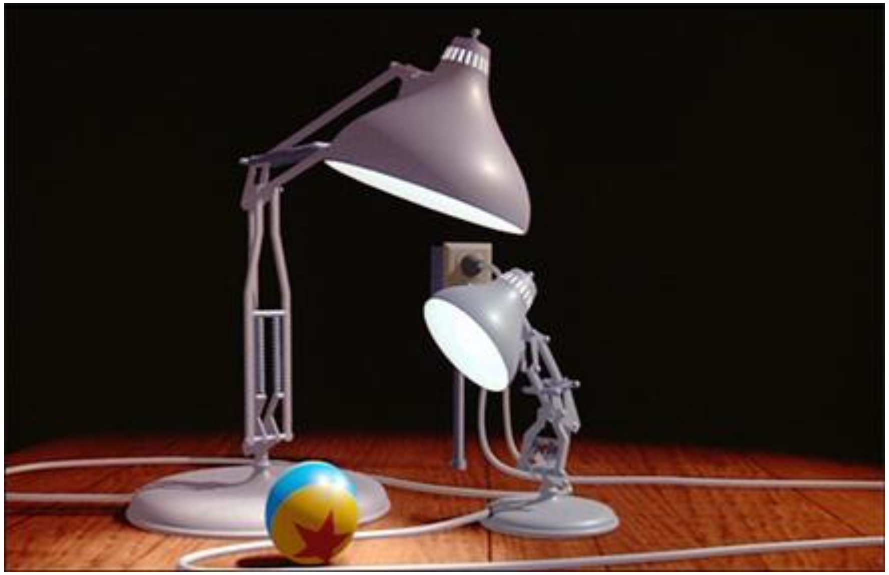
M&Ms: Freshmen Experience Processors

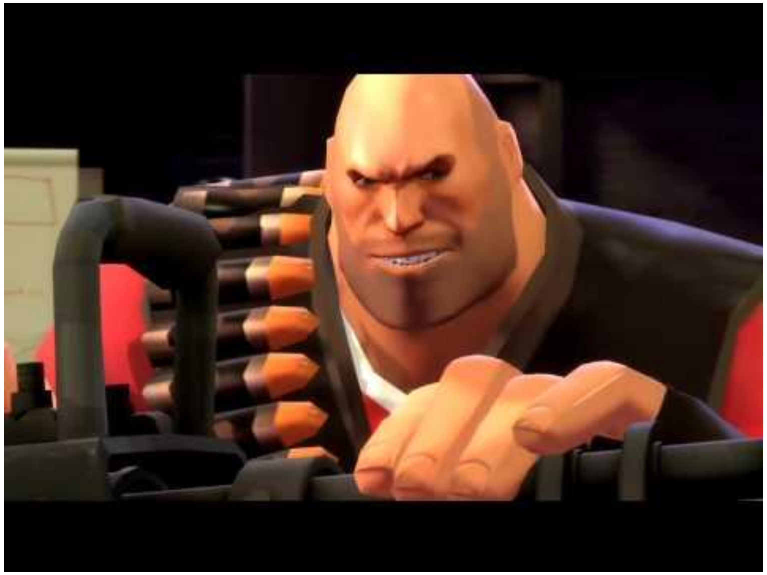
Michael Kazhdan

Thought Question

Which of these two videos was generated using ray-tracing?



Luxo Jr. (Pixar)



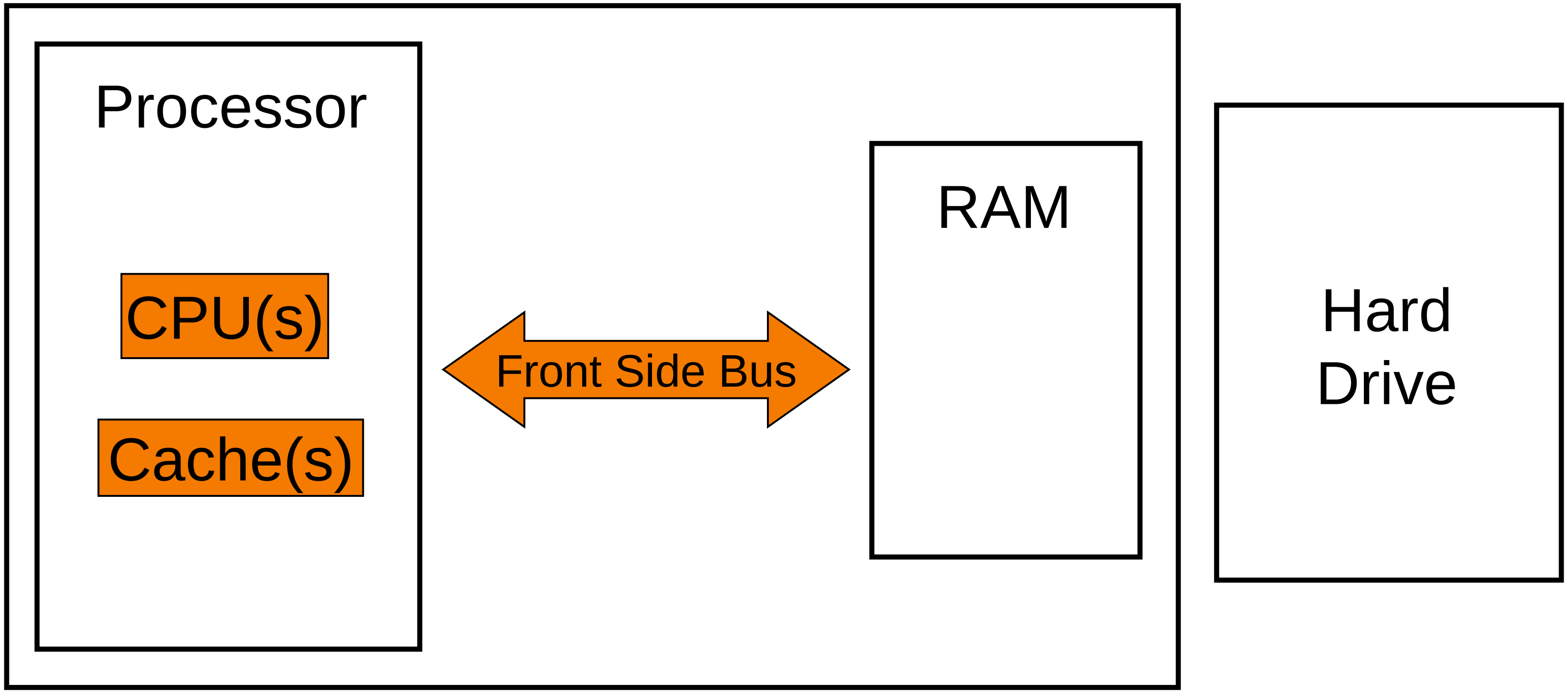
Meet the Heavy (Valve)

Introduction

Keywords:

- Clock Speed
- Cache
- RAM
- Front Side Bus Speed
- Good Coding

Hardware



How Fast Is Your Processor?

A standard measure of performance is the clock speed:

> “…the speed at which a microprocessor executes instructions…”

Examples:

- Pentium 4 670 Prescott: 3.8 GHz
- Athlon 64 3700+ Clawhammer: 2.4 GHz

How Fast Is Your Processor?

A standard measure of performance is the clock speed:

> "…the speed at which a microprocessor executes instructions…"

Examples:

Intel Core i-7 930 Bloomfield: 2.8 GHz (x4)

AMD Phenom II X4 965: 3.4 GHz (x4)

Does this mean that the AMD is a better processor than the Intel?

How Fast Is Your Processor?

A standard measure of performance is the clock speed:

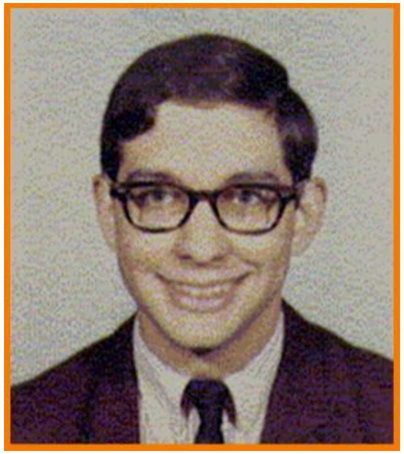
> "…the speed at which a microprocessor executes instructions…"

One question to consider is:

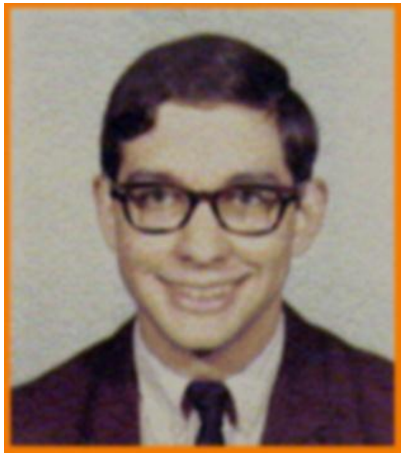
> How do processors break up operations?
>
> > More microprocessor instructions per operations usually means slower execution time per operation.

Example: Blurring an Image

- To blur an image, we create a new image where the value at a pixel is the average of the adjacent pixels.



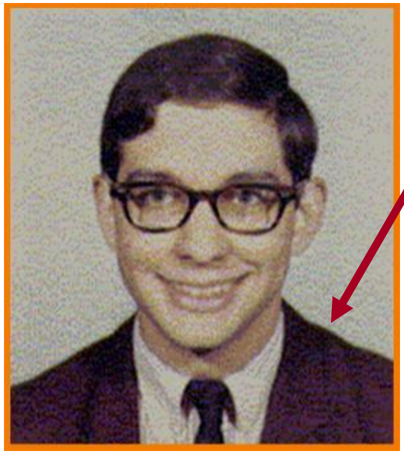
Original



Blur

Example: Blurring an Image

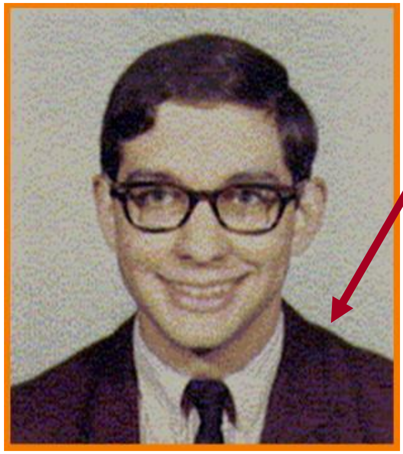
Pixel(x,y) = 36



Original

Example: Blurring an Image

Pixel(x,y) = 36



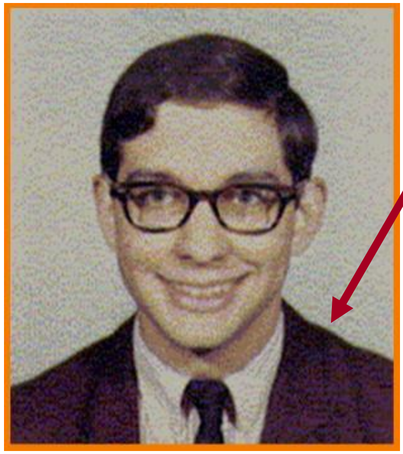
Original

	X - 1	X	X + 1
Y - 1	36	109	146
Y	32	36	109
Y + 1	32	36	73

Pixel(x,y) and its neighbors

Example: Blurring an Image

New value for Pixel(x,y) =
[36 + 109 + 146
32 + 36 + 109
32 + 36 + 73] / 9



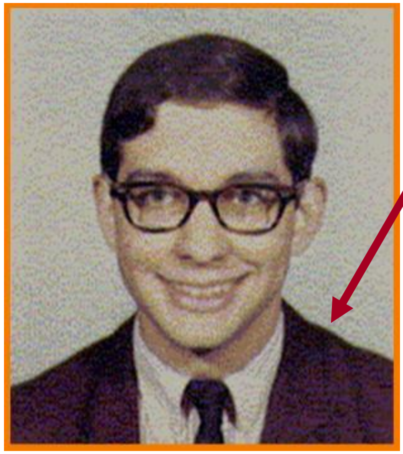
Original

	X - 1	X	X + 1
Y - 1	36	109	146
Y	32	36	109
Y + 1	32	36	73

Pixel(x,y) and its neighbors

Example: Blurring an Image

New value for Pixel(x,y) = 63



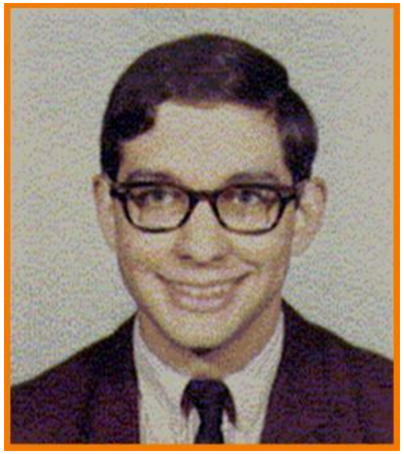
Original

	X - 1	X	X + 1
Y - 1	36	109	146
Y	32	36	109
Y + 1	32	36	73

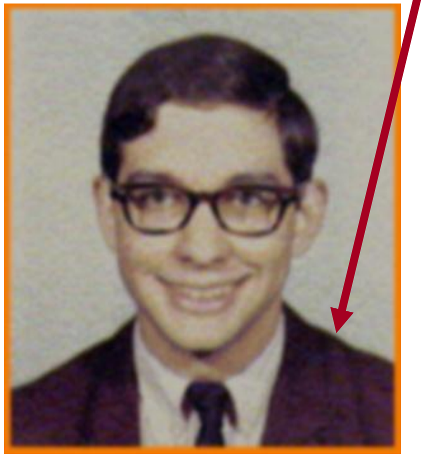
Pixel(x,y) and its neighbors

Example: Blurring an Image

New value for Pixel(x,y) = 63



Original



Blur

Example: Blurring an Image

Computational Complexity:

If the image is of size 100x100 the total number of computations we need to do is:

- [100 x 100] x 9 adds
- [100 x 100] x 1 divides

Example: Blurring an Image

Computational Complexity:

If the image is of size 100x100 the total number of computations we need to do is:

- [100 x 100] x 9 adds
- [100 x 100] x 1 divides

So the faster the processor, the more quickly it can do the blurring. Right?

Example: Blurring an Image

Computational Complexity:

If the image is of size 100x100 the total number of computations we need to do is:

- [100 x 100] x 9 adds
- [100 x 100] x 1 divides

So the faster the processor, the more quickly it can do the blurring. Right?

Wrong!

Example: Blurring an Image

In addition to processing the pixel values data, the processor also has to acquire the pixel data, which is stored on the hard drive.

Example: Blurring an Image

We could just go to the hard drive every time we need pixel information.

Example: Blurring an Image

We could just go to the hard drive every time we need pixel information.

[100 x 100] x 9 Hard Drive Queries

Example: Blurring an Image

We could just go to the hard drive every time we need pixel information.

[100 x 100] x 9 Hard Drive Queries

Problem: Getting information off the hard drive is slow, so our speed is determined by the speed of the hard drive communication, not the speed of the processor.

Example: Blurring an Image

Solution: We can use the fact that each pixel in the original, contributes to nine pixels in the blurred image.

Example: Blurring an Image

Solution: We can use the fact that each pixel in the original, contributes to nine pixels in the blurred image.

Read the entire image from the hard drive once and store it in temporary storage that is quicker to access.

Example: Blurring an Image

Solution: We can use the fact that each pixel in the original, contributes to nine pixels in the blurred image.

Read the entire image from the hard drive once and store it in temporary storage that is quicker to access.

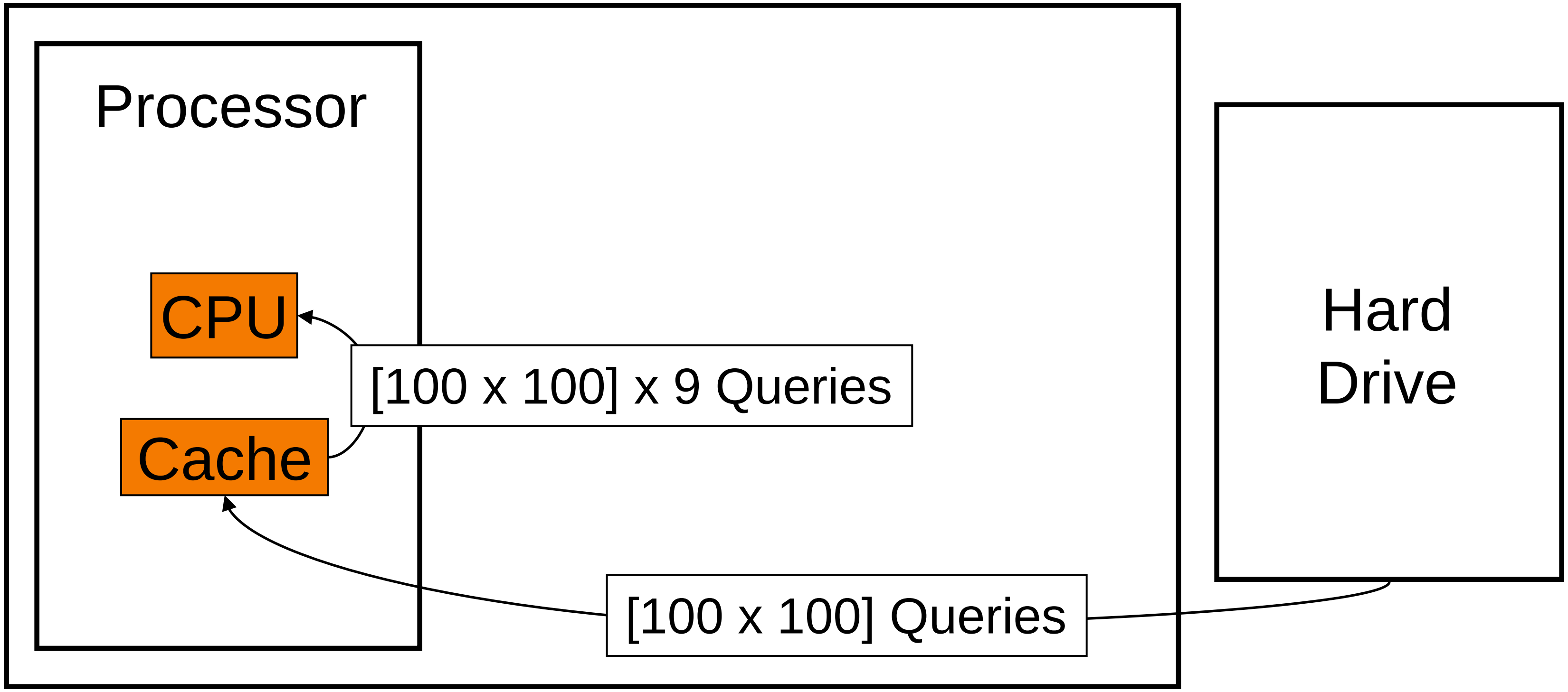
[100 x 100] Hard Drive Queries

[100 x 100] x 9 Temporary Storage Queries

Example: Blurring an Image

Cache:

This is memory on the processor that can be accessed very quickly.



Example: Blurring an Image

Cache:

If the entire image doesn’t fit into the cache, we can only load part of it.

This means that we have to swap in new memory from the disk to the cache when its not already in cache.

Typical caches sizes are about 512Kb.

A 500x500 color image is about 750Kb.

Example: Blurring an Image

Solution:

More Memory!

- Price: Fast memory is expensive.
- Space: There is only so much memory that can fit on the processor.

Example: Blurring an Image

RAM:

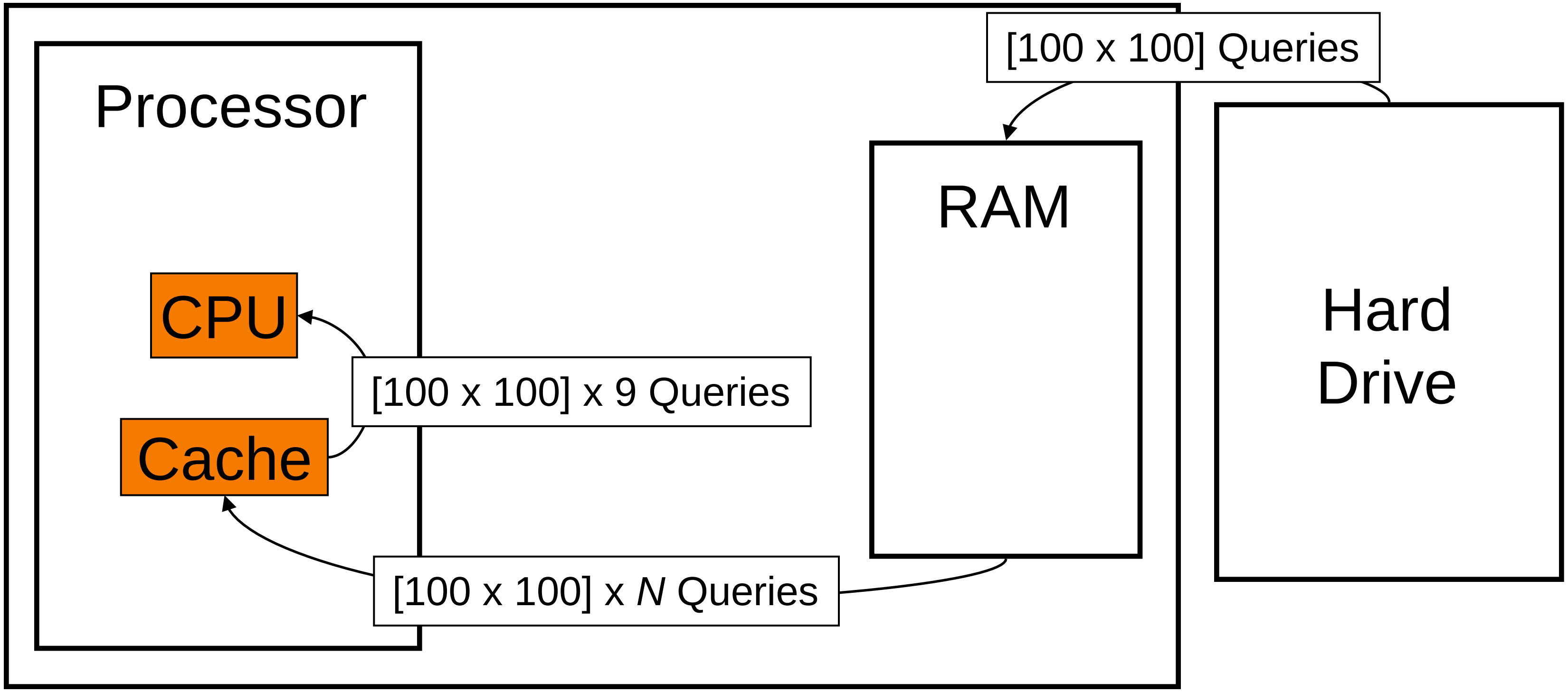
One solution is to have more memory between the cache and the hard drive:

- Can store more than the cache
- Is faster to access than the hard drive

Example: Blurring an Image

RAM:

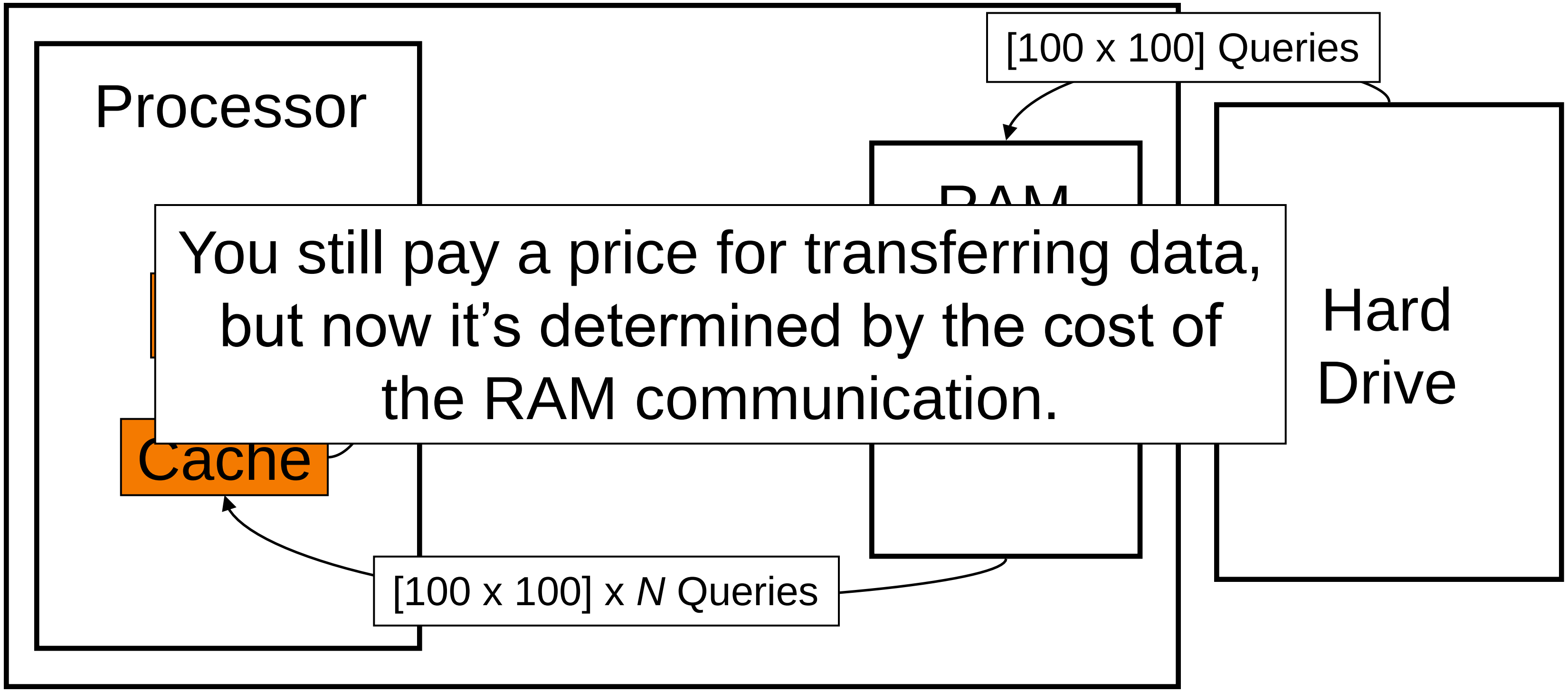
One solution is to have more memory between the cache and the hard drive:



Example: Blurring an Image

RAM:

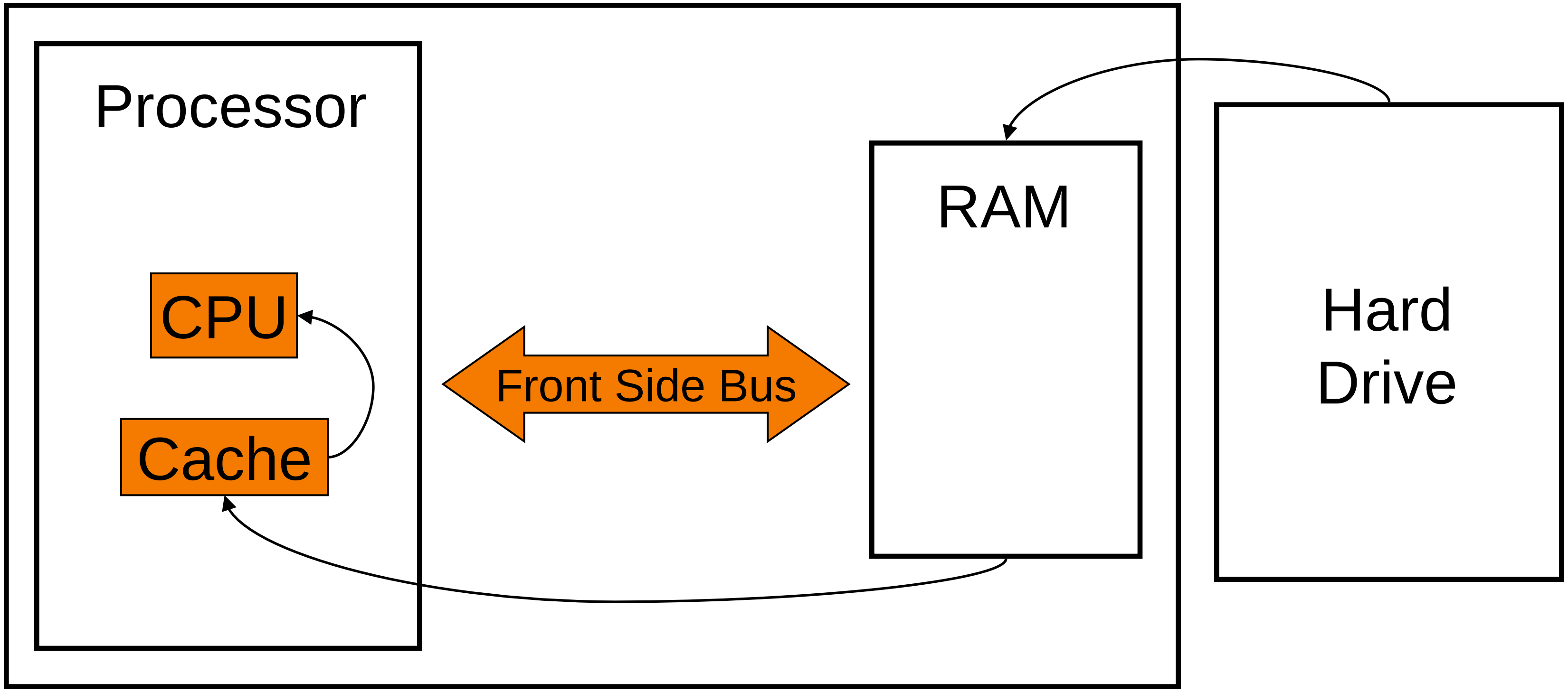
One solution is to have more memory between the cache and the hard drive:



You still pay a price for transferring data, but now it's determined by the cost of the RAM communication.

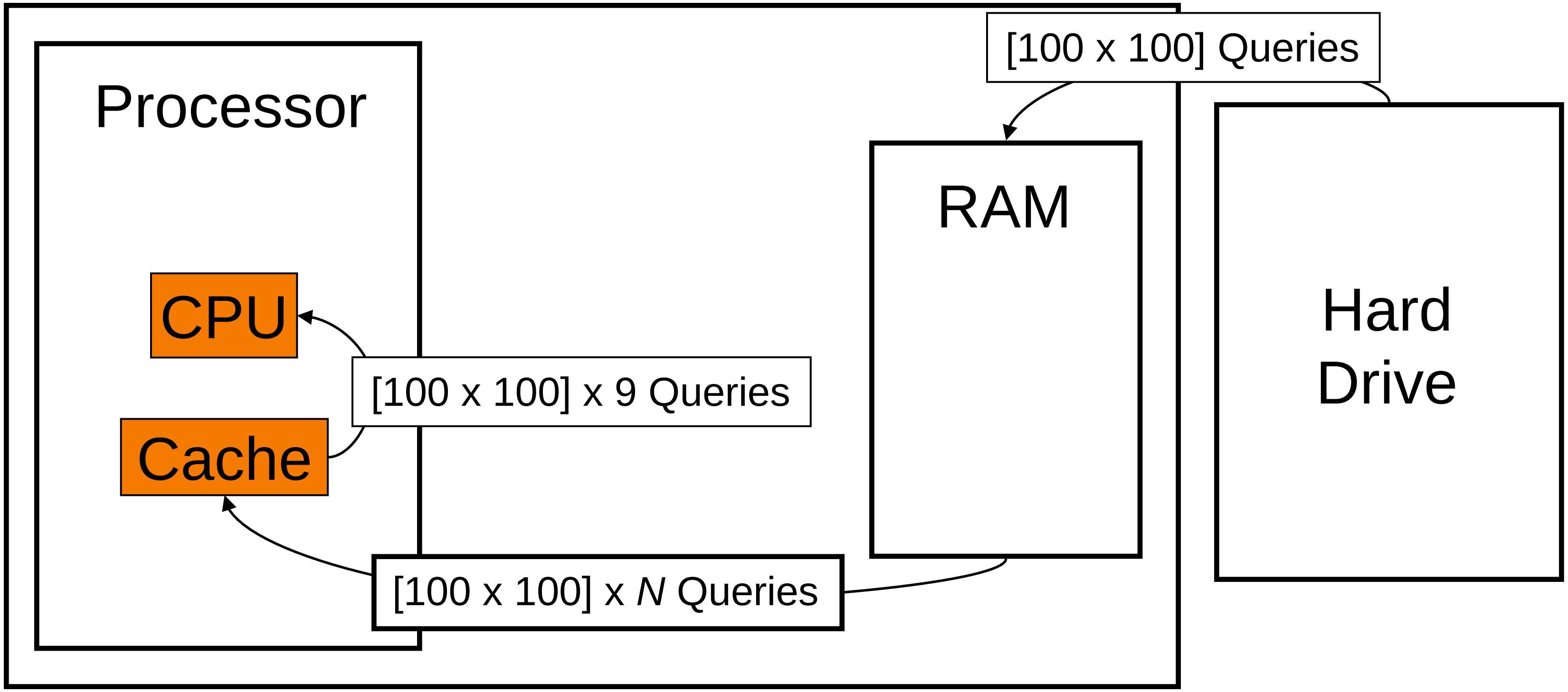
Example: Blurring an Image

RAM:
How quickly the processor “talks” to the memory is determined by the speed of the Front Side Bus (FSB).



Example: Blurring an Image

How many times do you transfer data from RAM to the processor?



Example: Blurring an Image

How many times do you transfer data from RAM to the processor?

This depends on two factors:

1. How big the cache is

Example: Blurring an Image

How many times do you transfer data from RAM to the processor?

This depends on two factors:

1. How big the cache is
2. How good your code is

Example: Blurring an Image

Code Quality:

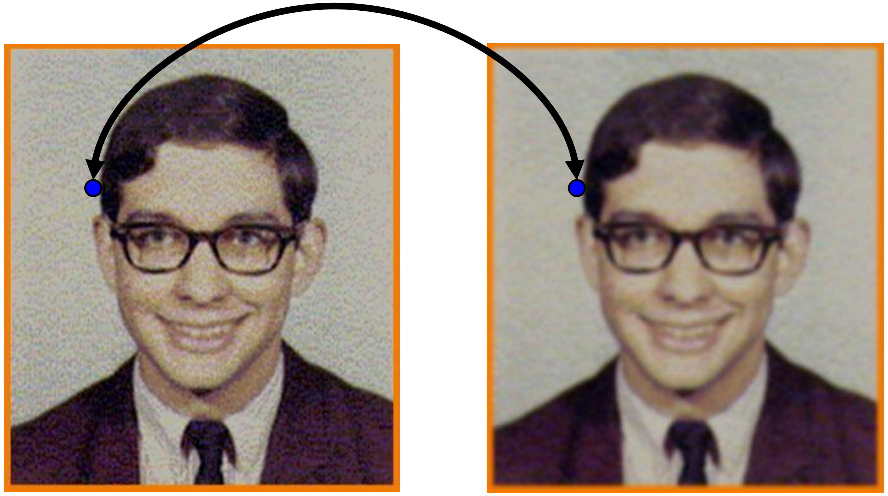
Remember that we only have to load memory from RAM into the cache when it’s not already there.

One goal of writing good code is to try to ensure that data accesses are contiguous.

Example: Blurring an Image

Example of Bad Coding:

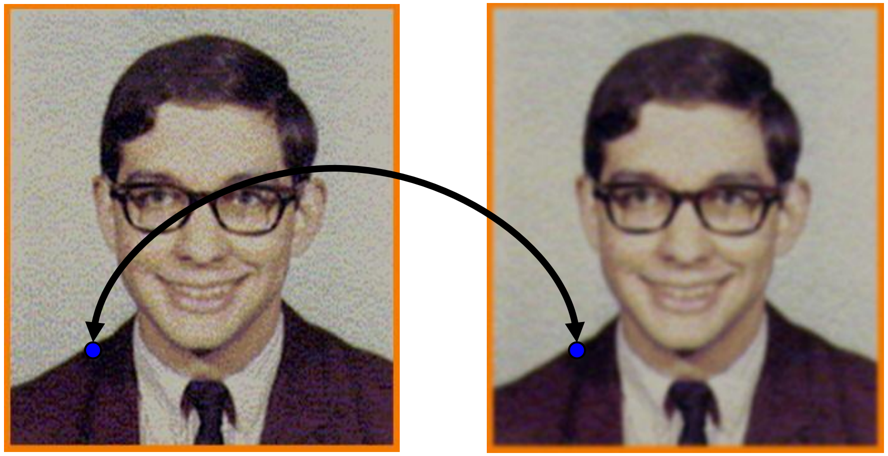
Data accesses are random



Example: Blurring an Image

Example of Bad Coding:

Data accesses are random



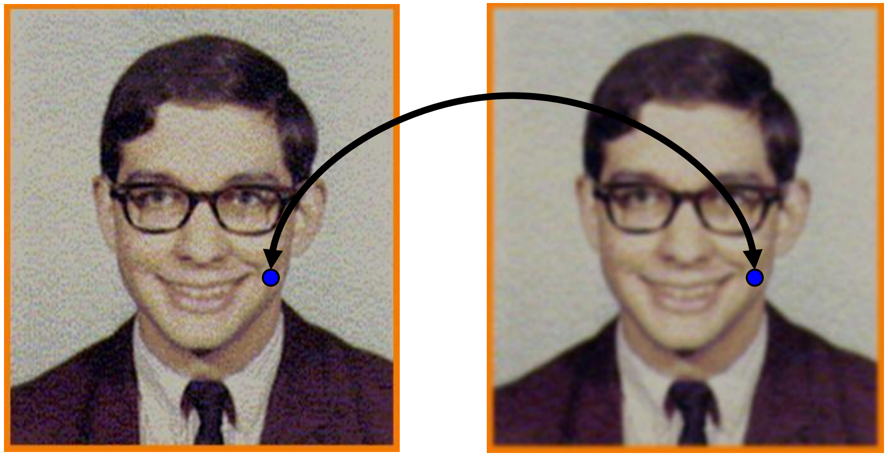
Original

Blur

Example: Blurring an Image

Example of Bad Coding:

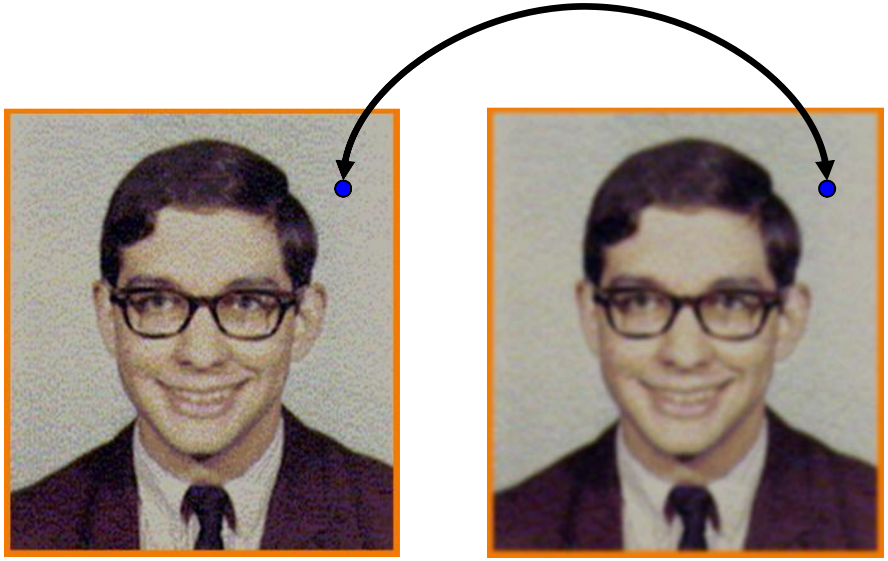
Data accesses are random



Example: Blurring an Image

Example of Bad Coding:

Data accesses are random



Original

Blur

Example: Blurring an Image

Example of Bad Coding:

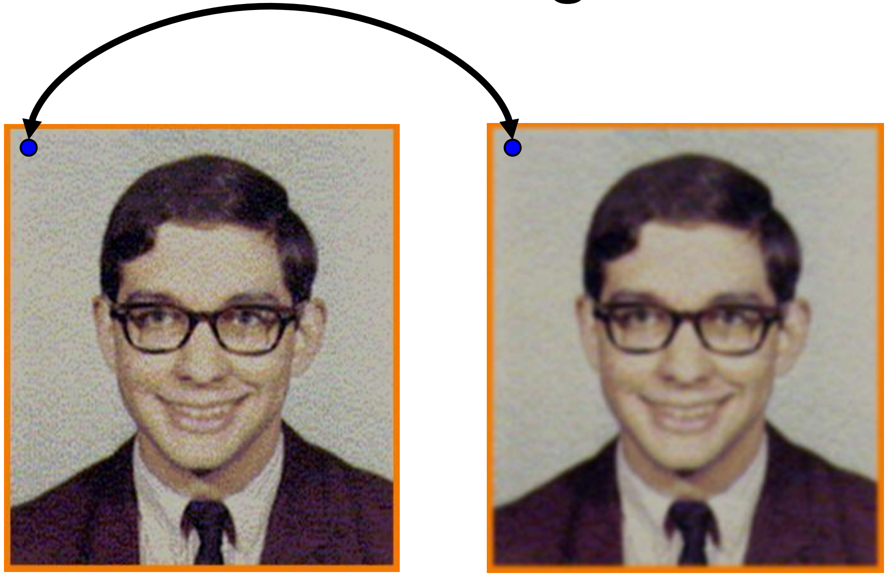
Data accesses are random.

By the time we need to re-use a pixel value, it is probably out of cache because we have had to load other stuff in the meantime.

Example: Blurring an Image

Example of Good Coding:

Data accesses are contiguous



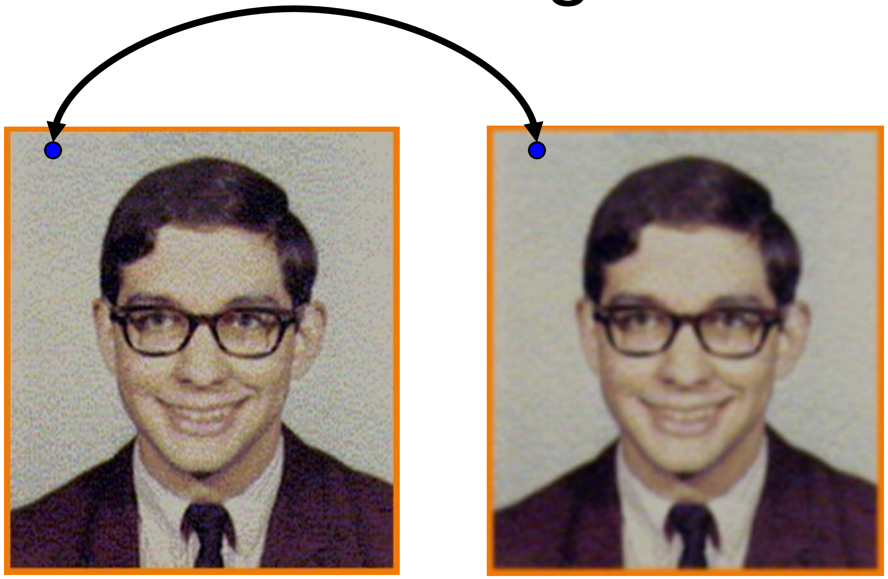
Original

Blur

Example: Blurring an Image

Example of Good Coding:

Data accesses are contiguous



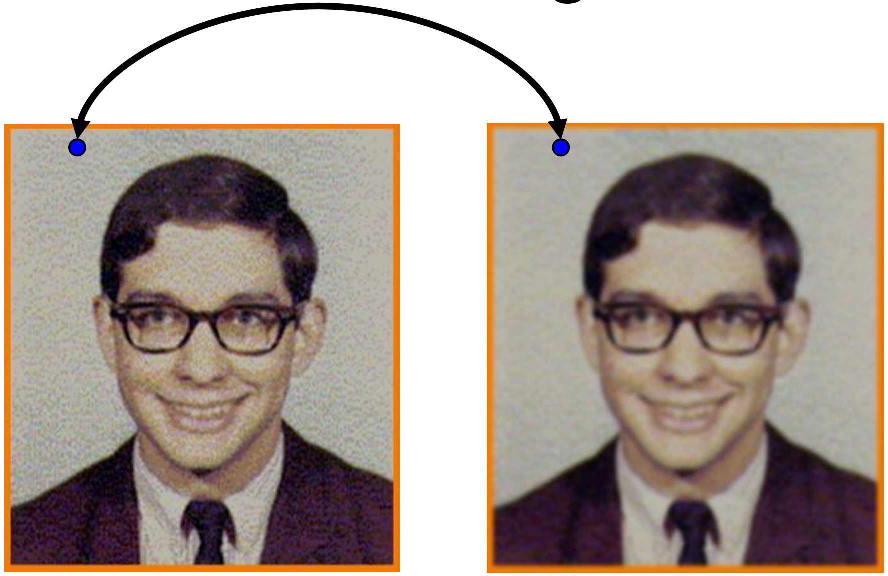
Original

Blur

Example: Blurring an Image

Example of Good Coding:

Data accesses are contiguous



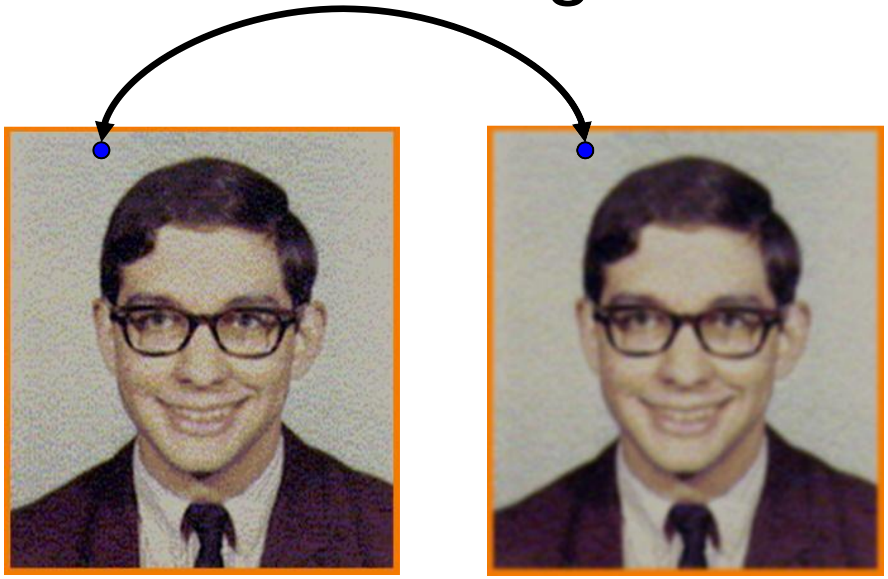
Original

Blur

Example: Blurring an Image

Example of Good Coding:

Data accesses are contiguous



Original

Blur

Example: Blurring an Image

Example of Good Coding:

Data accesses are contiguous.

We try to reuse the cache data as much as possible before we load in new data into the cache.