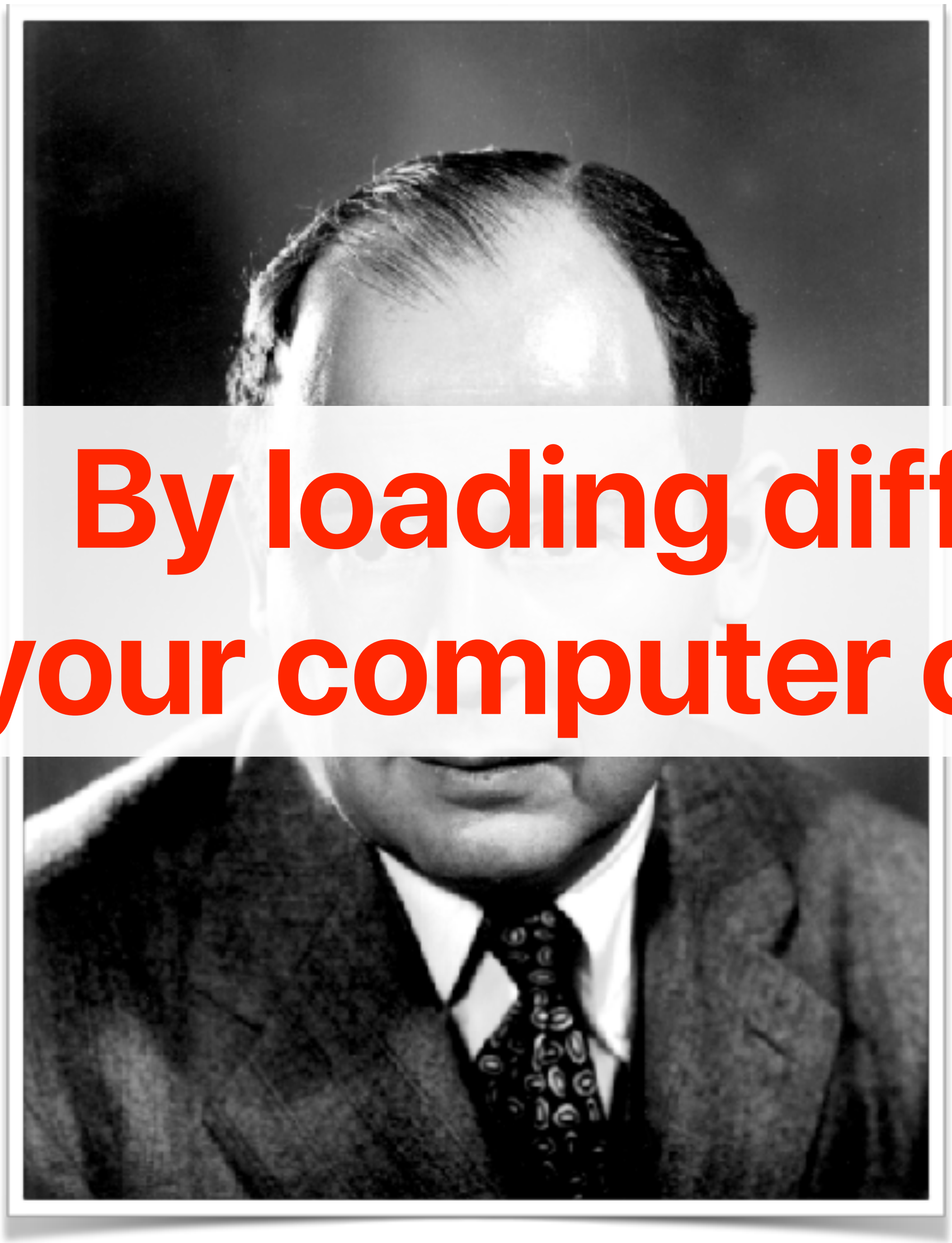
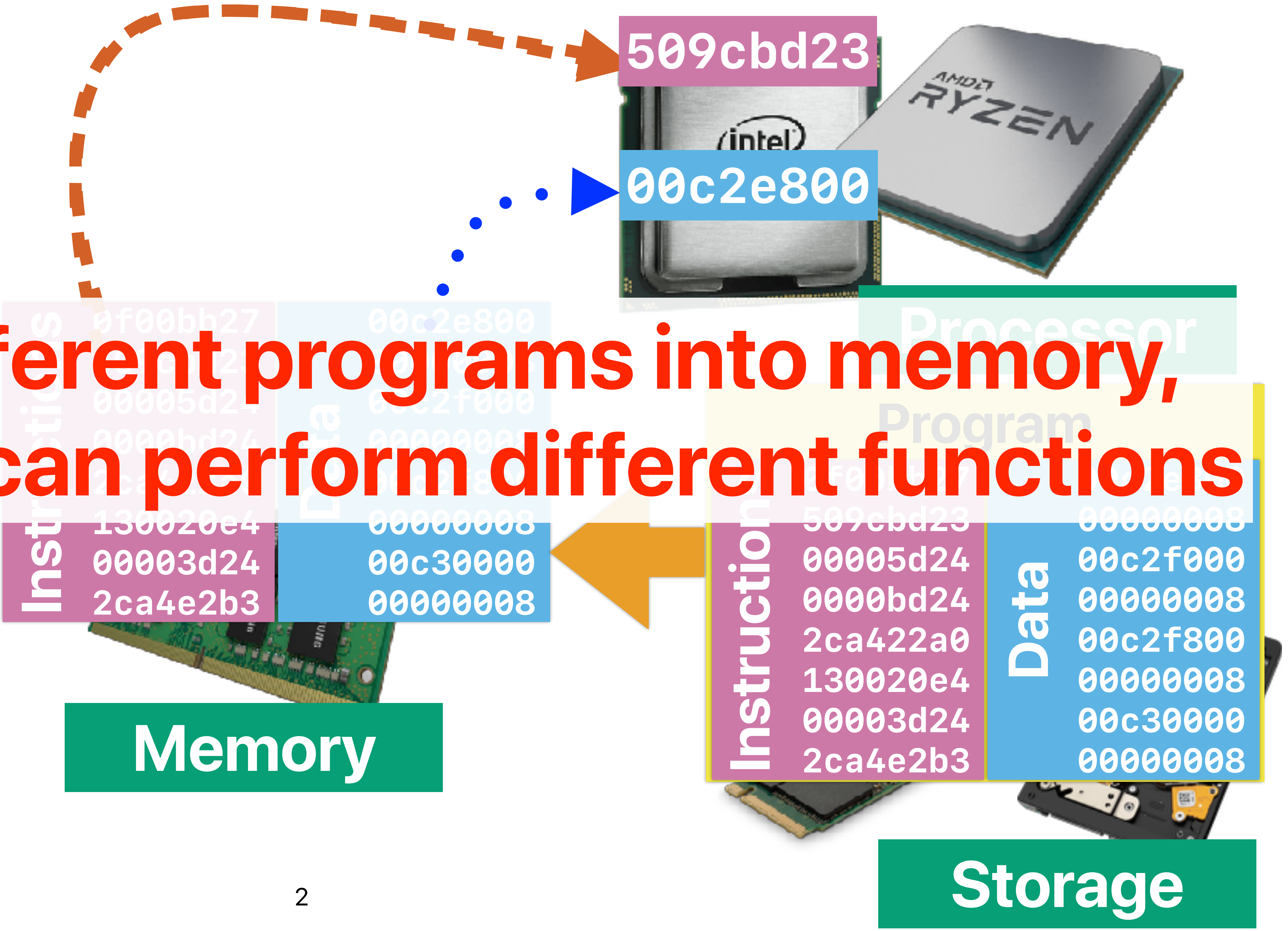
Performance (II)

Hung-Wei Tseng

Recap: von Neumman Architecture





By loading different programs into memory, your computer can perform different functions

Recap: Definition of "Performance"

$$Performance = \frac{1}{Execution\ Time}$$

$$Execution\ Time = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

$$ET = IC \times CPI \times CT$$

$$1GHz = 10^9 Hz = \frac{1}{10^9} sec\ per\ cycle = 1\ ns\ per\ cycle$$

$$\frac{1}{Frequency(i.e., clock\ rate)}$$

Recap: Definition of "Speedup"

- The relative performance between two machines, X and Y. Y is n times faster than X

$$n = \frac{Execution\ Time_X}{Execution\ Time_Y}$$

- The speedup of Y over X

$$Speedup = \frac{Execution\ Time_X}{Execution\ Time_Y}$$

Recap: demo — programmer & performance

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
  for(j = 0; j < ARRAY_SIZE; j++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
  for(i = 0; i < ARRAY_SIZE; i++)
  {
    c[i][j] = a[i][j]+b[i][j];
  }
}
```

A		B
$O(n^2)$	**Complexity**	$O(n^2)$
Same	**Instruction Count?**	**Same**
Same	**Clock Rate**	**Same**
Better	**CPI**	**Worse**

Recap: How programmer affects performance?

- Performance equation consists of the following three factors
  1. ✓ IC
  2. ✓ CPI
  3. ✓ CT

  How many can a **programmer** affect?

  A. 0
  B. 1
  C. 2
  [✓ D. 3]

Recap: programming languages & performance

- How many instructions are there in “Hello, world!”

	Instruction count	LOC	Ranking
C	600k	6	1
C++	3M	6	2
Java	~210M	8	5
Perl	10M	4	3
Python	~30M	1	4

Recap: demo revisited — compiler optimization

- Compiler can reduce the instruction count, change CPI — with “limited scope”
- Compiler CANNOT help improving “crummy” source code

```
    if(option)
        std::sort(data, data + arraySize);
```

Compiler can never add this — only the programmer can!

```
    for (unsigned c = 0; c < arraySize*1000; ++c) {
            if (data[c%arraySize] >= INT_MAX/2)
                sum ++;
        }
    }
```

Summary of CPU Performance Equation

$$Performance = \frac{1}{Execution\ Time}$$

$$Execution\ Time = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

$$ET = IC \times CPI \times CT$$

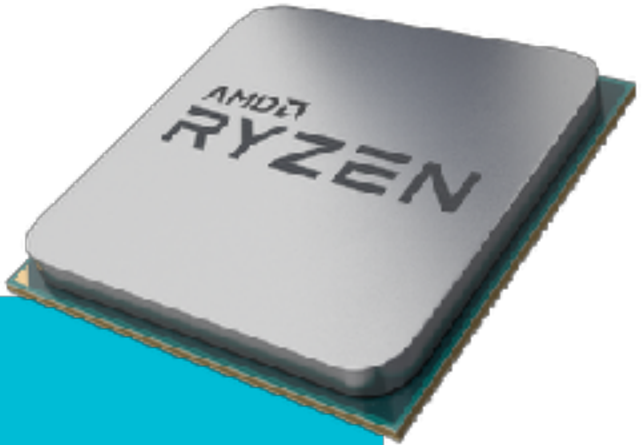
- IC (Instruction Count)
  - ISA, Compiler, algorithm, programming language, **programmer**
- CPI (Cycles Per Instruction)
  - Machine Implementation, microarchitecture, compiler, application, algorithm, programming language, **programmer**
- Cycle Time (Seconds Per Cycle)
  - Process Technology, microarchitecture, **programmer**

Instruction Set Architecture (ISA) & Performance

Recap: ISA — the interface b/w processor/software

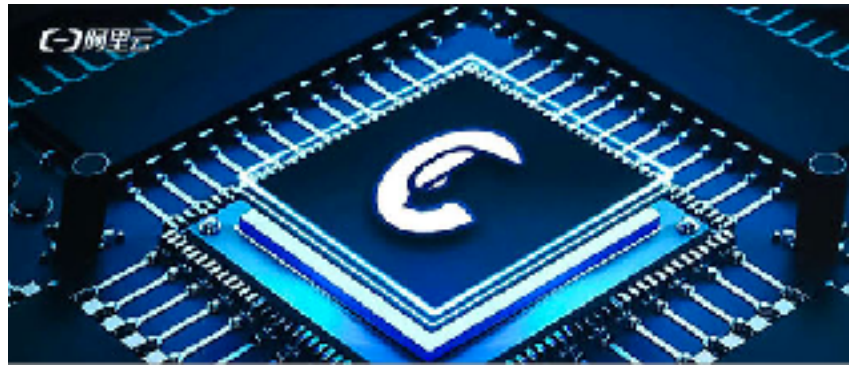
- Operations
  - Arithmetic/Logical, memory access, control-flow (e.g., branch, function calls)
  - Operands
    - Types of operands — register, constant, memory addresses
    - Sizes of operands — byte, 16-bit, 32-bit, 64-bit
- Memory space
  - The size of memory that programs can use
  - The addressing of each memory locations
  - The modes to represent those addresses

Popular ISAs



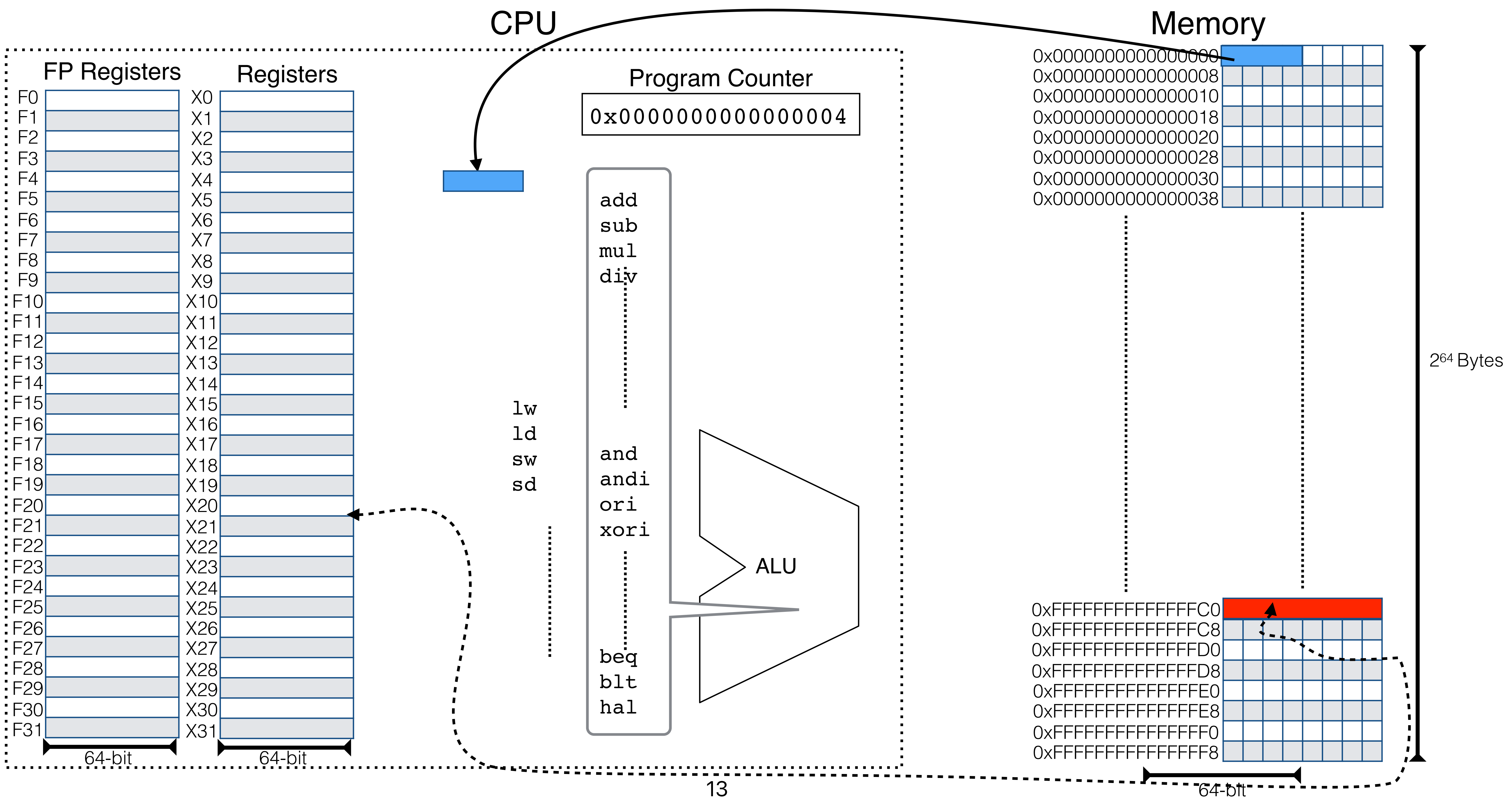
x86

arm



RISC-V

The abstracted "RISC-V" machine



Outline

- Definition of “Performance”
- What affects each factor in “Performance Equation”
- Instruction Set Architecture & Performance

Subset of RISC-V instructions

Category	Instruction	Usage	Meaning
Arithmetic	add	add x1, x2, x3	x1 = x2 + x3
	addi	addi x1,x2, 20	x1 = x2 + 20
	sub	sub x1, x2, x3	x1 = x2 − x3
Logical	and	and x1, x2, x3	x1 = x2 & x3
	or	or x1, x2, x3	x1 = x2 | x3
	andi	andi x1, x2, 20	x1 = x2 & 20
	sll	sll x1, x2, 10	x1 = x2 ∗ 2^10
	srl	srl x1, x2, 10	x1 = x2 / 2^10
Data Transfer	ld	ld x1, 8(x2)	x1 = mem[x2+8]
	sd	sd x1, 8(x2)	mem[x2+8] = x1
Branch	beq	beq x1, x2, **25**	if(x1 == x2), PC = PC + **100**
	bne	bne x1, x2, **25**	if(x1 != x2), PC = PC + **100**
Jump	jal	jal **25**	$ra = PC **+ 4**, PC = 100
	jr	jr $ra	PC = $ra

The only type of instructions can access memory

Popular ISAs

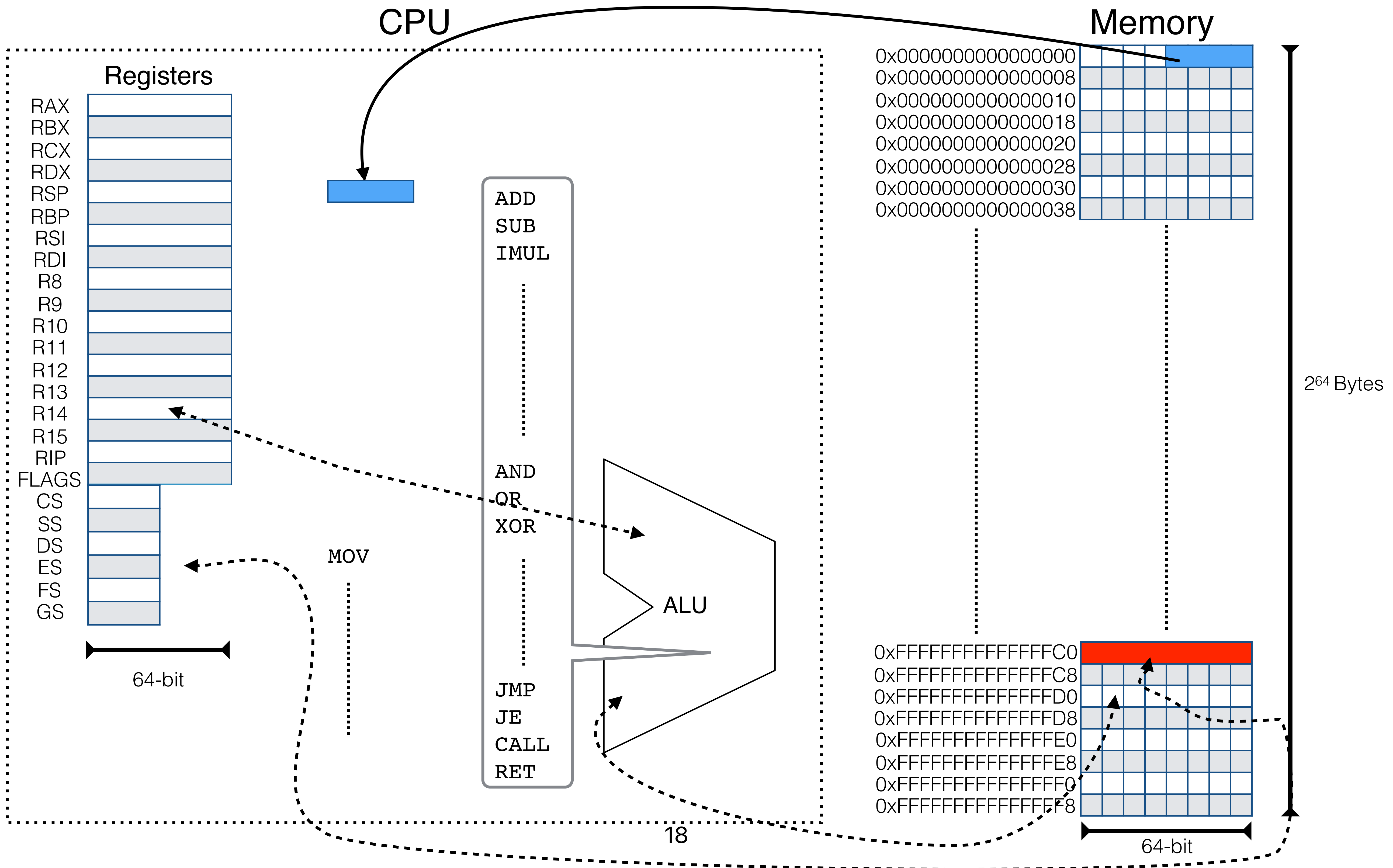
Complex Instruction Set Computers (CISC)

Reduced Instruction Set Computers (RISC)

How many operations: CISC v.s. RISC

- CISC (Complex Instruction Set Computing)
  - Examples: x86, Motorola 68K
  - Provide **many** **powerful/complex** instructions
    - Many: more than 1503 instructions since 2016
    - Powerful/complex: an instruction can perform both ALU and memory operations
    - Each instruction takes more cycles to execute
- RISC (Reduced Instruction Set Computer)
  - Examples: ARMv8, RISC-V, MIPS (the first RISC instruction, invented by the authors of our textbook)
  - Each instruction only performs simple tasks
  - Easy to decode
  - Each instruction takes less cycles to execute

The abstracted x86 machine



RISC-V v.s. x86

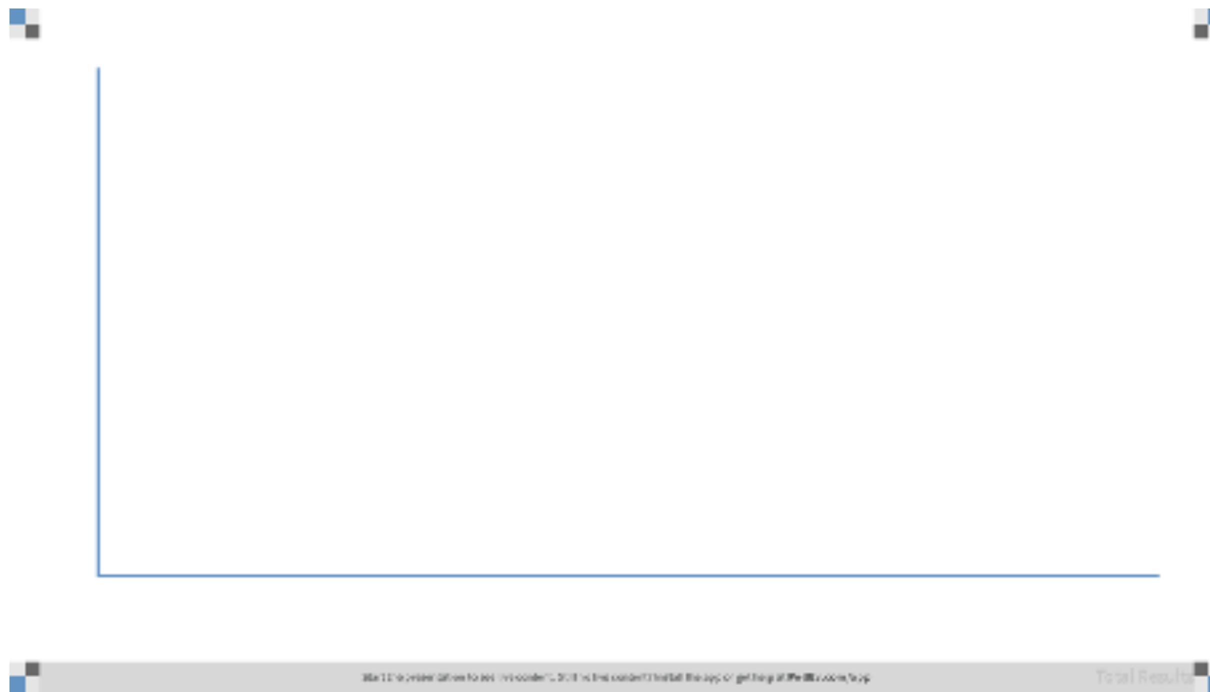
	RISC-V	x86
ISA type	Reduced Instruction Set Computers (RISC)	Complex Instruction Set Computers (CISC)
instruction width	32 bits	1 ~ 17 bytes
code size	larger	smaller
registers	32	16
addressing modes	reg+offset	base+offset base+index scaled+index scaled+index+offset
hardware	simple	complex

https://www.pollev.com/hungweitseng close in 01:00

RISC-V v.s. x86

- Using the same language, the same source code, regarding the compiled program on x86 and RISC-V, how many of the following statements is/are "generally" correct?
  - ① The RISC-V version would contain more instructions than its x86 version
  - ② The RISC-V version tends to incur fewer memory accesses than its x86 version
  - ③ The RISC-V version needs a processor with higher clock rate than its x86 version if the CPI of both versions are similar
  - ④ The RISC-V version needs a processor with lower CPI than its x86 version if the x86 processor runs at the same clock rate

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4



https://www.pollev.com/hungweitseng close in 01:00

RISC-V v.s. x86

- Using the same language, the same source code, regarding the compiled program on x86 and RISC-V, how many of the following statements is/are "generally" correct?
  - ✓ The RISC-V version would contain more instructions than its x86 version
  - ✓ The RISC-V version tends to incur fewer memory accesses than its x86 version
  - ✓ The RISC-V version needs a processor with higher clock rate than its x86 version if the CPI of both versions are similar
  - ✓ The RISC-V version needs a processor with lower CPI than its x86 version if the x86 processor runs at the same clock rate

A. 0
B. 1
C. 2
D. 3
[✓ E. 4]

Amdahl’s Law — and It’s Implication in the Multicore Era

H&P Chapter 1.9

M. D. Hill and M. R. Marty. Amdahl’s Law in the Multicore Era. In Computer, vol. 41, no. 7, pp. 33-38, July 2008.

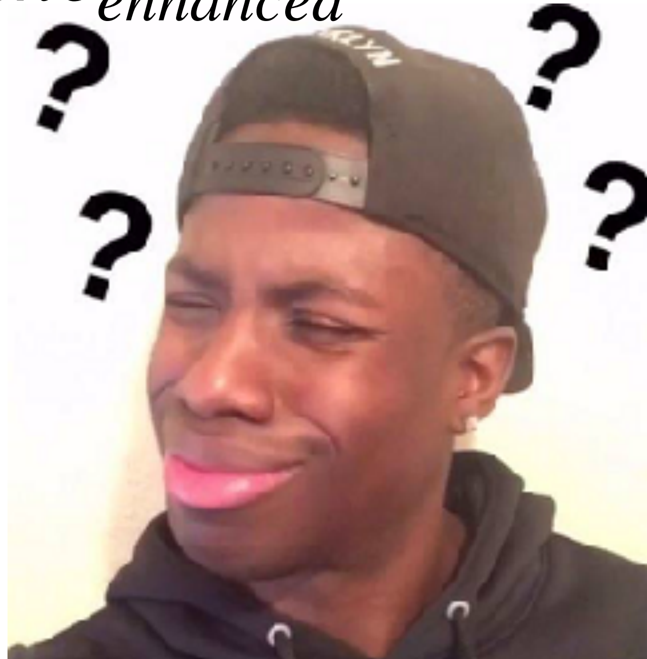
Amdahl's Law

$$Speedup_{enhanced}(f, s) = \frac{1}{(1 - f) + \frac{f}{s}}$$

f — The fraction of time in the original program

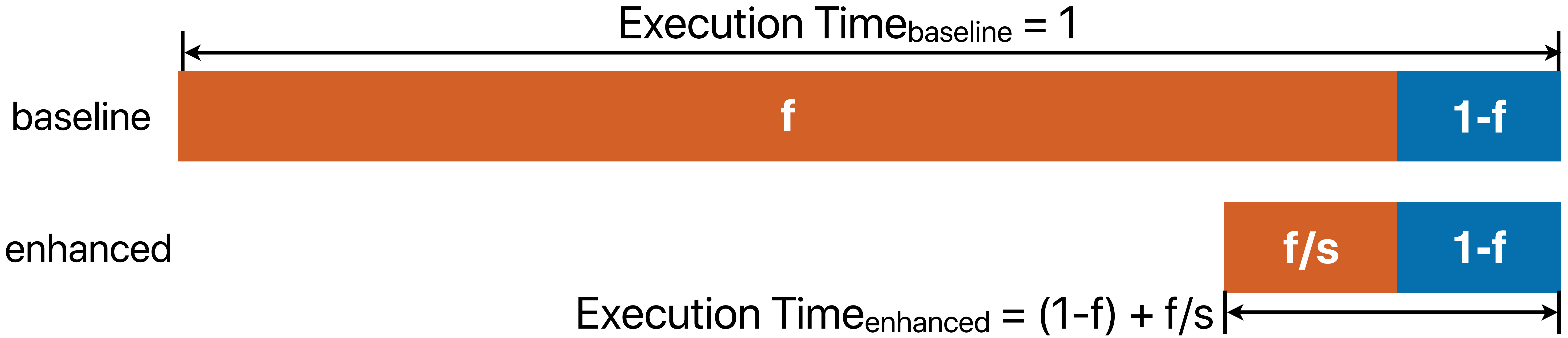
s — The speedup we can achieve on f

$$Speedup_{enhanced} = \frac{Execution\ Time_{baseline}}{Execution\ Time_{enhanced}}$$



Amdahl's Law

$$Speedup_{enhanced}(f, s) = \frac{1}{(1-f) + \frac{f}{s}}$$


Execution Time_{baseline} = 1
baseline
f
1-f
enhanced
f/s
1-f
Execution Time_{enhanced} = (1-f) + f/s

$$Speedup_{enhanced} = \frac{Execution\ Time_{baseline}}{Execution\ Time_{enhanced}} = \frac{1}{(1-f) + \frac{f}{s}}$$

https://www.pollev.com/hungweitseng close in 01:00

Recap: Speedup

- Assume that we have an application composed with a total of 500000 instructions, in which 20% of them are the load/store instructions with an average CPI of 6 cycles, and the rest instructions are integer instructions with average CPI of 1 cycle when using a 2GHz processor.
  - If we double the CPU clock rate to 4GHz that helps to accelerate all instructions by 2x except that load/store instruction cannot be improved — their CPI will become 12 cycles. What's the performance improvement after this change?
  - No change
  - 1.25
  - 1.5
  - 2
  - None of the above

Recap: Speedup

- Assume that we have an application composed with a total of 500000 instructions, in which 20% of them are the load/store instructions with an average CPI of 6 cycles, and the rest instructions are integer instructions with average CPI of 1 cycle when using a 2GHz processor.
  - If we double the CPU clock rate to 4GHz that helps to accelerate all instructions by 2x except that load/store instruction cannot be improved — their CPI will become 12 cycles. What's the performance improvement after this change?
  - No change
  - 1.25
  - 1.5
  - 2
  - None of the above

Recap: Speedup

- Assume that we have an application composed with a total of 500000 instructions, in which 20% of them are the load/store instructions with an average CPI of 6 cycles, and the rest instructions are integer instructions with average CPI of 1 cycle when using a 2GHz processor.
  - If we double the CPU clock rate to 4GHz that helps to accelerate all instructions by 2x except that load/store instruction cannot be improved — their CPI will become 12 cycles. What's the performance improvement after this change?
  - No change
  - $\boxed{1.25}$
  - 1.5
  - 2
  - None of the above

$$ET = IC \times CPI \times CT$$

$$ET_{baseline} = (5 \times 10^5) \times (20\% \times 6 + 80\% \times 1) \times \frac{1}{2 \times 10^{-9}} sec = 5^{-3}$$

$$ET_{enhanced} = (5 \times 10^5) \times (20\% \times 12 + 80\% \times 1) \times \frac{1}{4 \times 10^{-9}} sec = 4^{-3}$$

$$Speedup = \frac{Execution\ Time_{baseline}}{Execution\ Time_{enhanced}}$$

$$= \frac{5}{4} = 1.25$$

Replay using Amdahl's Law

- Assume that we have an application composed with a total of 500000 instructions, in which 20% of them are the load/store instructions with an average CPI of 6 cycles, and the rest instructions are integer instructions with average CPI of 1 cycle when using a 2GHz processor.
  - If we double the CPU clock rate to 4GHz that helps to accelerate all instructions by 2x except that load/store instruction cannot be improved — their CPI will become 12 cycles. What's the performance improvement after this change?

How much time in load/store? $500000 \times (0.2 \times 6) \times 0.5\ ns = 300000\ ns \rightarrow 60\,\%$

How much time in the rest? $500000 \times (0.8 \times 1) \times 0.5\ ns = 200000\ ns \rightarrow 40\,\%$

$$Speedup_{enhanced}(f, s) = \frac{1}{(1-f) + \frac{f}{s}}$$

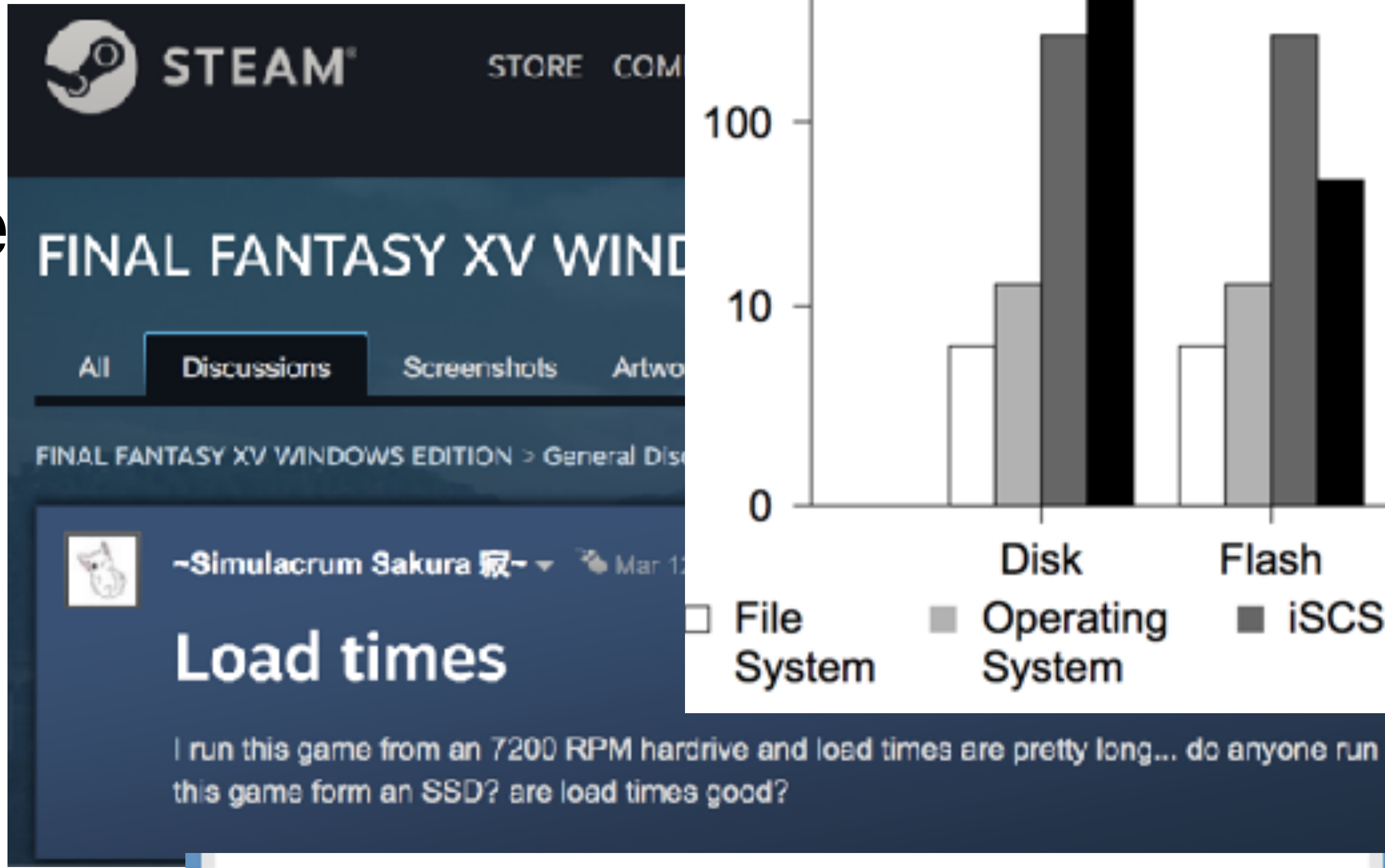
$$Speedup_{enhanced}(40\,\%, 2) = \frac{1}{(1-40\%) + \frac{40\,\%}{2}} = 1.25 \times$$

FINAL FANTASY XIV

https://www.pollev.com/hungweitseng close in 01:00

Practicing Amdahl's Law

- Final Fantasy XV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time. By how much can we speed up the map loading process?
  - A. ~7x
  - B. ~10x
  - C. ~17x
  - D. ~29x
  - E. ~100x


STEAM STORE COM

FINAL FANTASY XV WIND

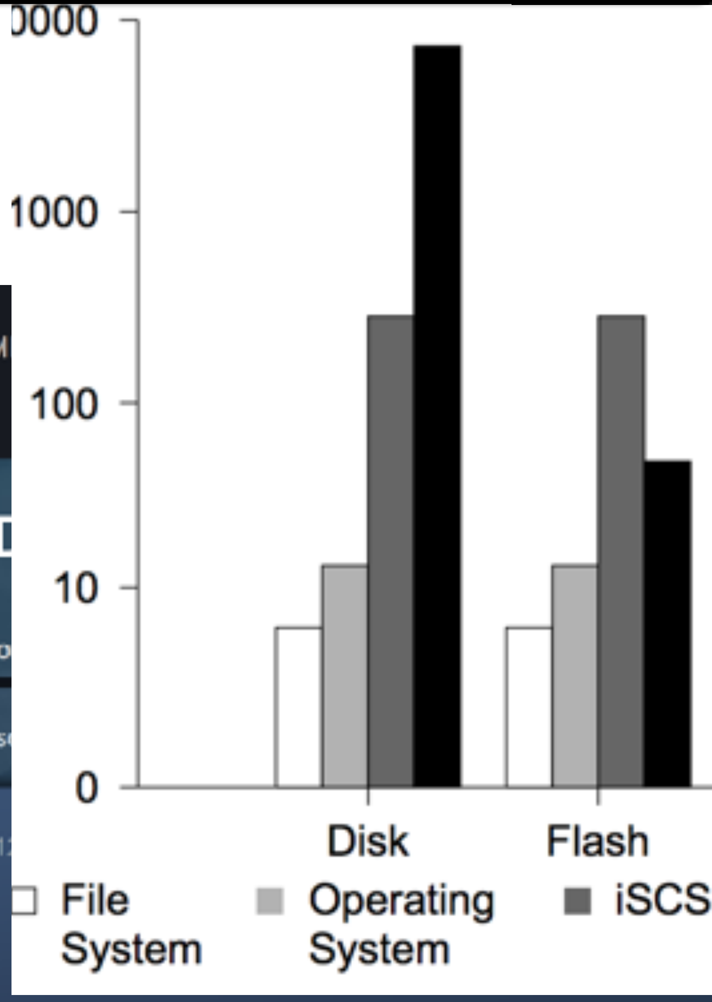
All Discussions Screenshots Artwo

FINAL FANTASY XV WINDOWS EDITION > General Dis

~Simulacrum Sakura 寂~ Mar 1

Load times

I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



https://www.pollev.com/hungweitseng close in 01:00

Practicing Amdahl's Law

- Final Fantasy XV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time. By how much can we speed up the map loading process?
  A. ~7x
  B. ~10x
  [✓ C. ~17x]
  D. ~29x
  E. ~100x

$$Speedup_{enhanced}(95\%, 100) = \frac{1}{(1-95\%)+\frac{95\%}{100}} = 16.81\times$$

STEAM STORE COM

FINAL FANTASY XV WIND

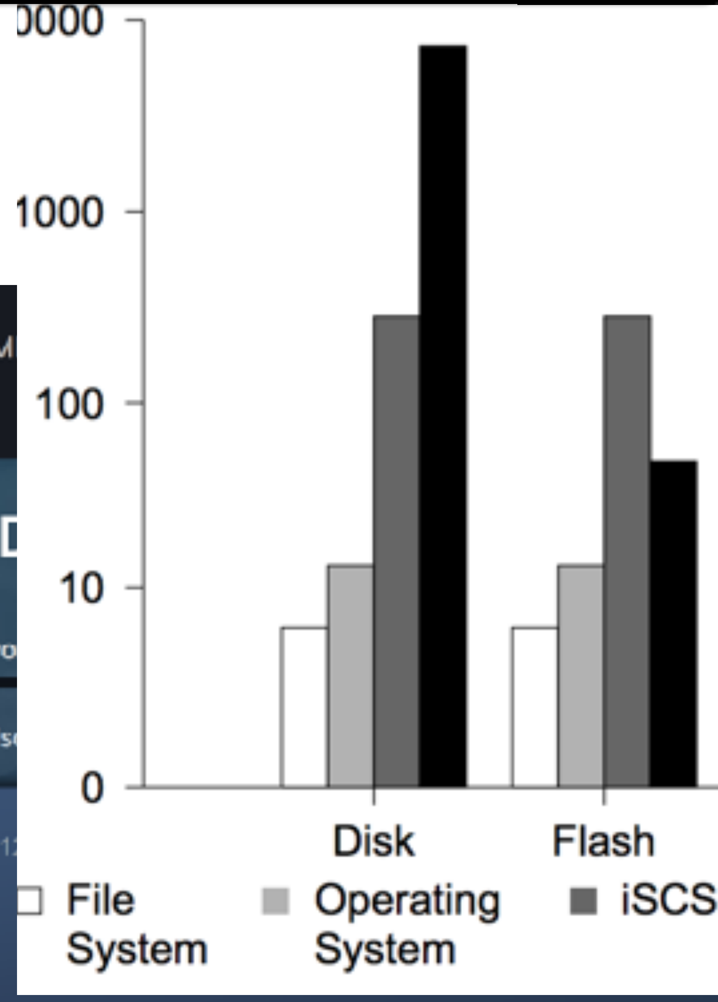
All | Discussions | Screenshots | Artwo

FINAL FANTASY XV WINDOWS EDITION > General Dis

~Simulacrum Sakura 冢~ Mar 1

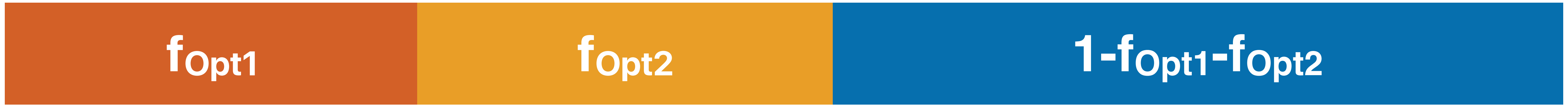
Load times

I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



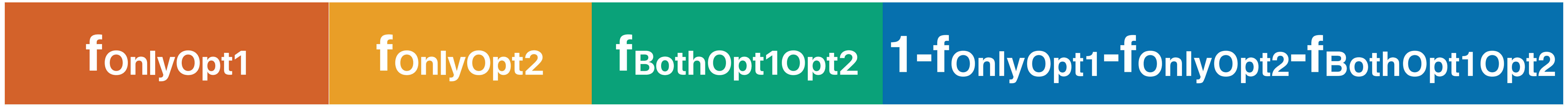
Amdahl's Law on Multiple Optimizations

- We can apply Amdahl's law for multiple optimizations
- These optimizations must be dis-joint!
  - If optimization #1 and optimization #2 are dis-joint:



$$Speedup_{enhanced}(f_{Opt1}, f_{Opt2}, s_{Opt1}, s_{Opt2}) = \frac{1}{(1 - f_{Opt1} - f_{Opt2}) + \frac{f\_Opt1}{s\_Opt1} + \frac{f\_Opt2}{s\_Opt2}}$$

  - If optimization #1 and optimization #2 are not dis-joint:



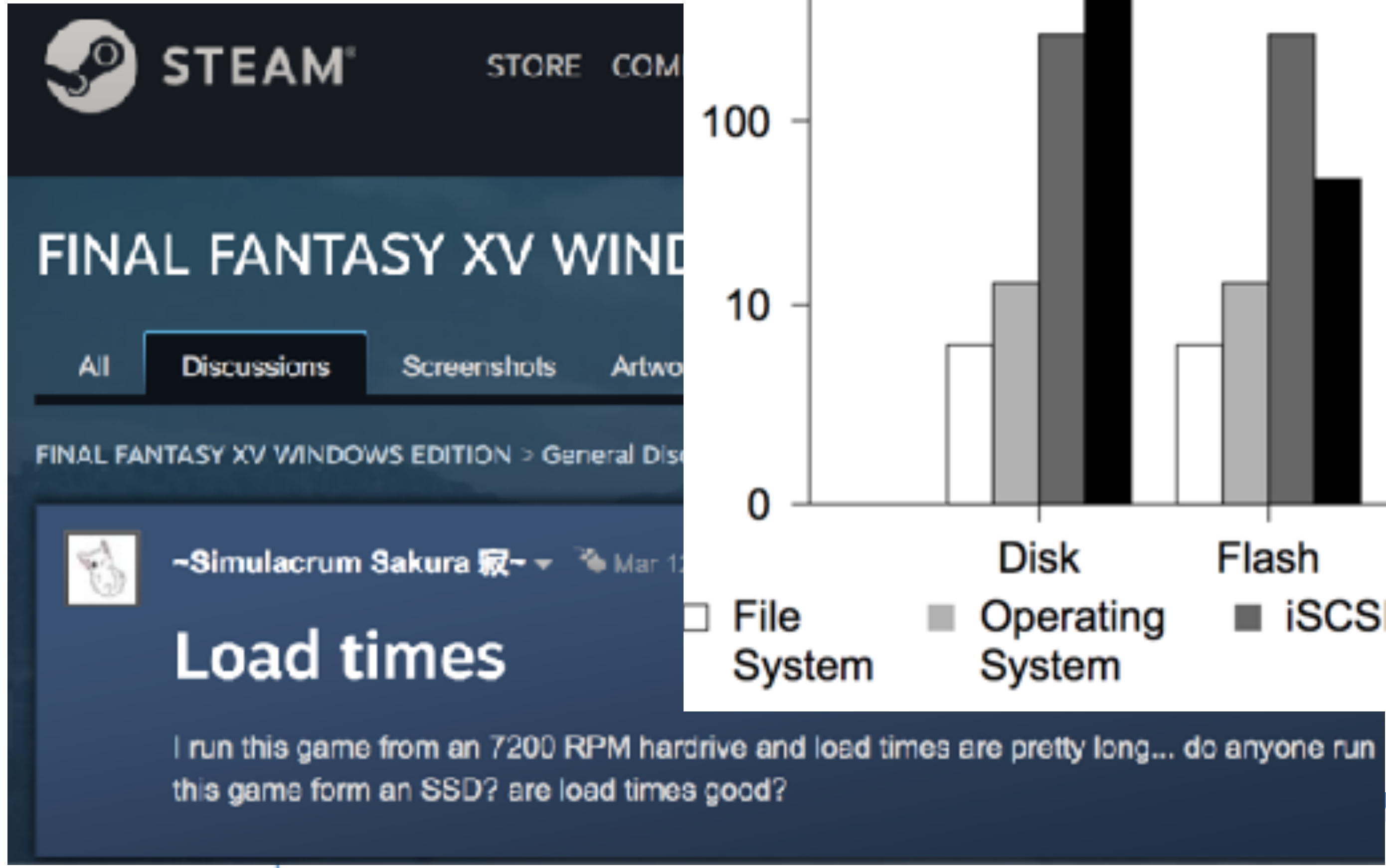
$f_{OnlyOpt1}$
$f_{OnlyOpt2}$
$f_{BothOpt1Opt2}$
$1\text{-}f_{OnlyOpt1}\text{-}f_{OnlyOpt2}\text{-}f_{BothOpt1Opt2}$

$$Speedup_{enhanced}(f_{OnlyOpt1}, f_{OnlyOpt2}, f_{BothOpt1Opt2}, s_{OnlyOpt1}, s_{OnlyOpt2}, s_{BothOpt1Opt2})$$
$$= \frac{1}{(1 - f_{OnlyOpt1} - f_{OnlyOpt2} - f_{BothOpt1Opt2}) + + \frac{f\_BothOpt1Opt2}{s\_BothOpt1Opt2} + \frac{f\_OnlyOpt1}{s\_OnlyOpt1} + \frac{f\_OnlyOpt2}{s\_OnlyOpt2}}$$

https://www.pollev.com/hungweitseng close in 01:00

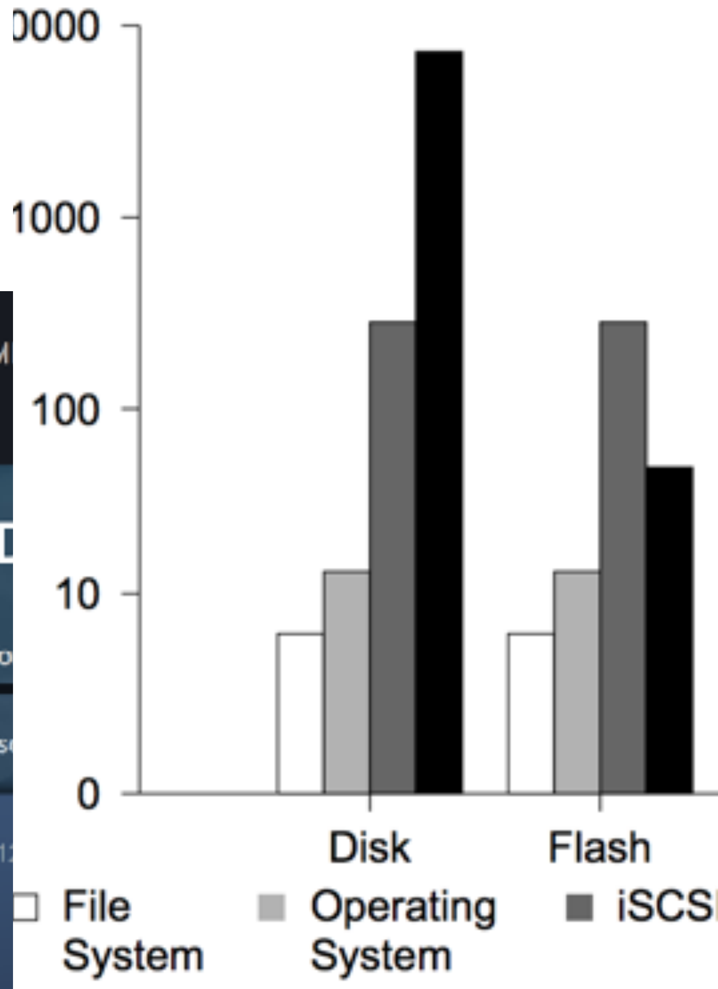
Practicing Amdahl's Law (2)

- Final Fantasy XIV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time and a better processor to accelerate the software overhead by 2x. By how much can we speed up the map loading process?
  A. ~7x
  B. ~10x
  C. ~17x
  D. ~29x
  E. ~100x



Load times

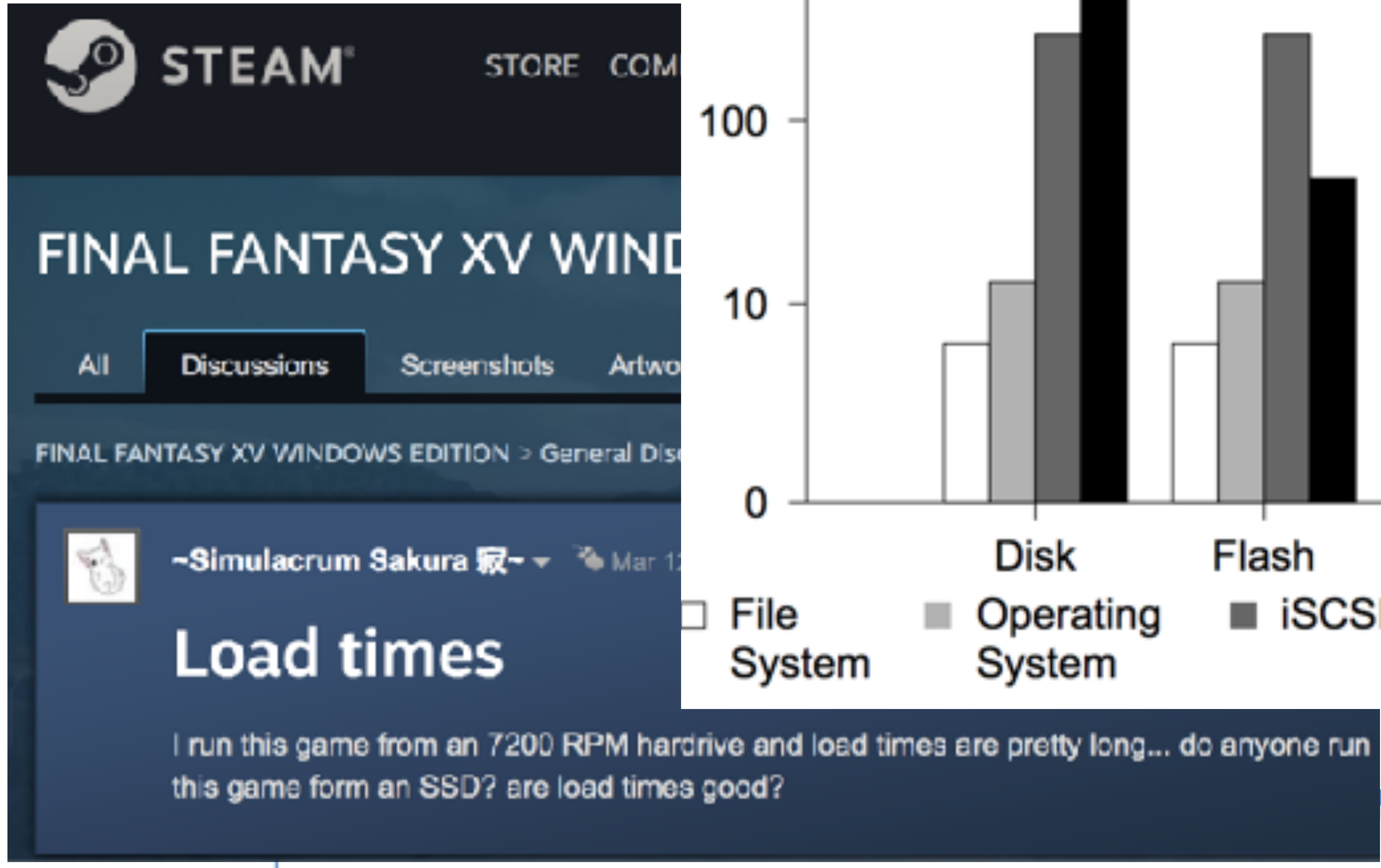
I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



https://www.pollev.com/hungweitseng close in 01:00

Practicing Amdahl's Law (2)

- Final Fantasy XIV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time and a better processor to accelerate the software overhead by 2x. By how much can we speed up the map loading process?
    A. ~7x
    B. ~10x
    C. ~17x
    D. ~29x
    E. ~100x



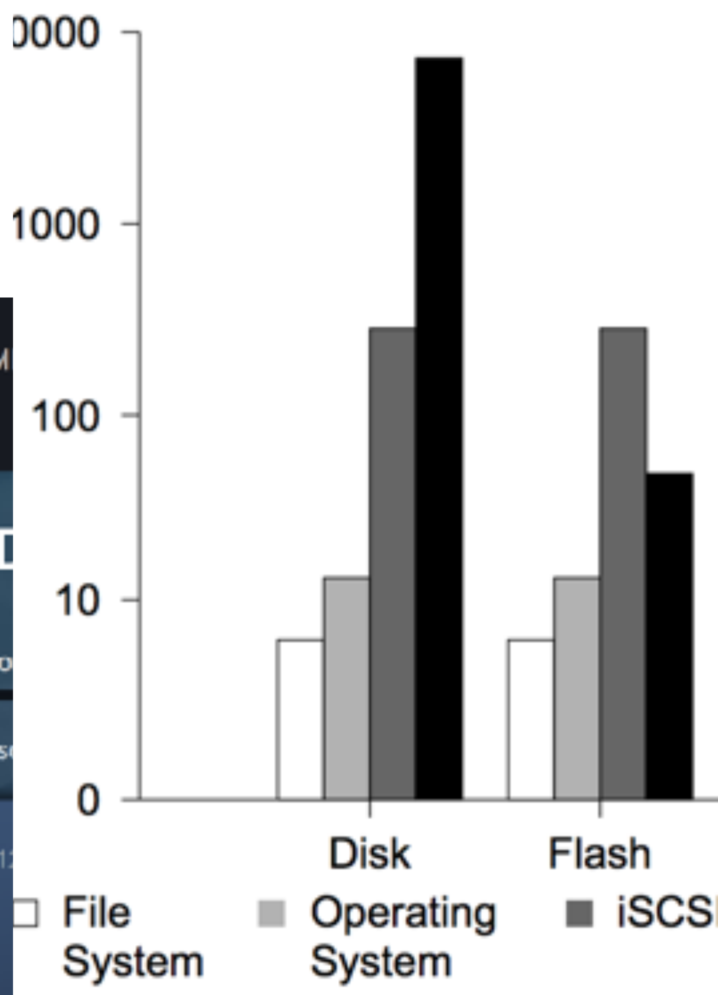
FINAL FANTASY XV WIND

FINAL FANTASY XV WINDOWS EDITION > General Dis

~Simulacrum Sakura 寂~ Mar 1

Load times

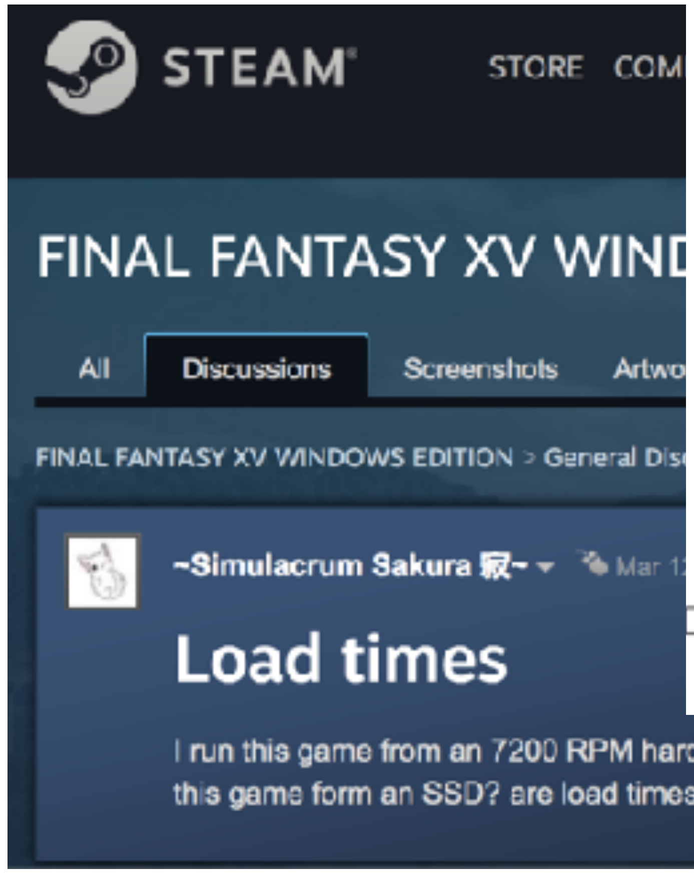
I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



Practicing Amdahl's Law (2)

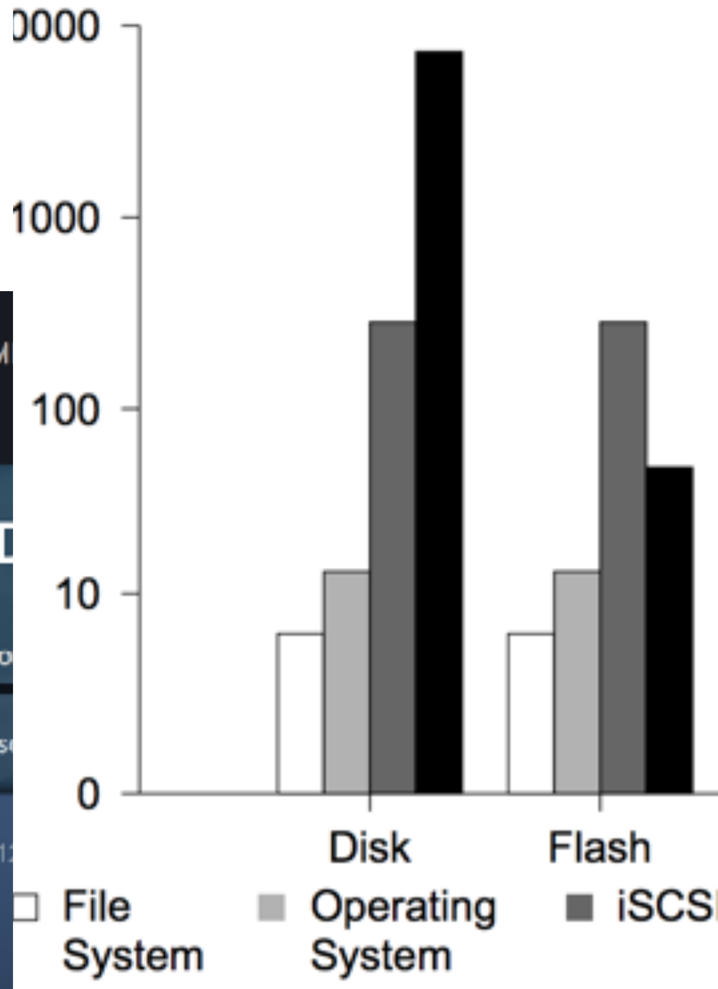
- Final Fantasy XIV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time and a better processor to accelerate the software overhead by 2x. By how much can we speed up the map loading process?
  - A. ~7x
  - B. ~10x
  - C. ~17x
  - [✓ D. ~29x]
  - E. ~100x

$$Speedup_{enhanced}(95\%, 5\%, 100, 2) = \frac{1}{(1 - 95\% - 5\%) + \frac{95\%}{100} + \frac{5\%}{2}} = 28.98\times$$



Load times

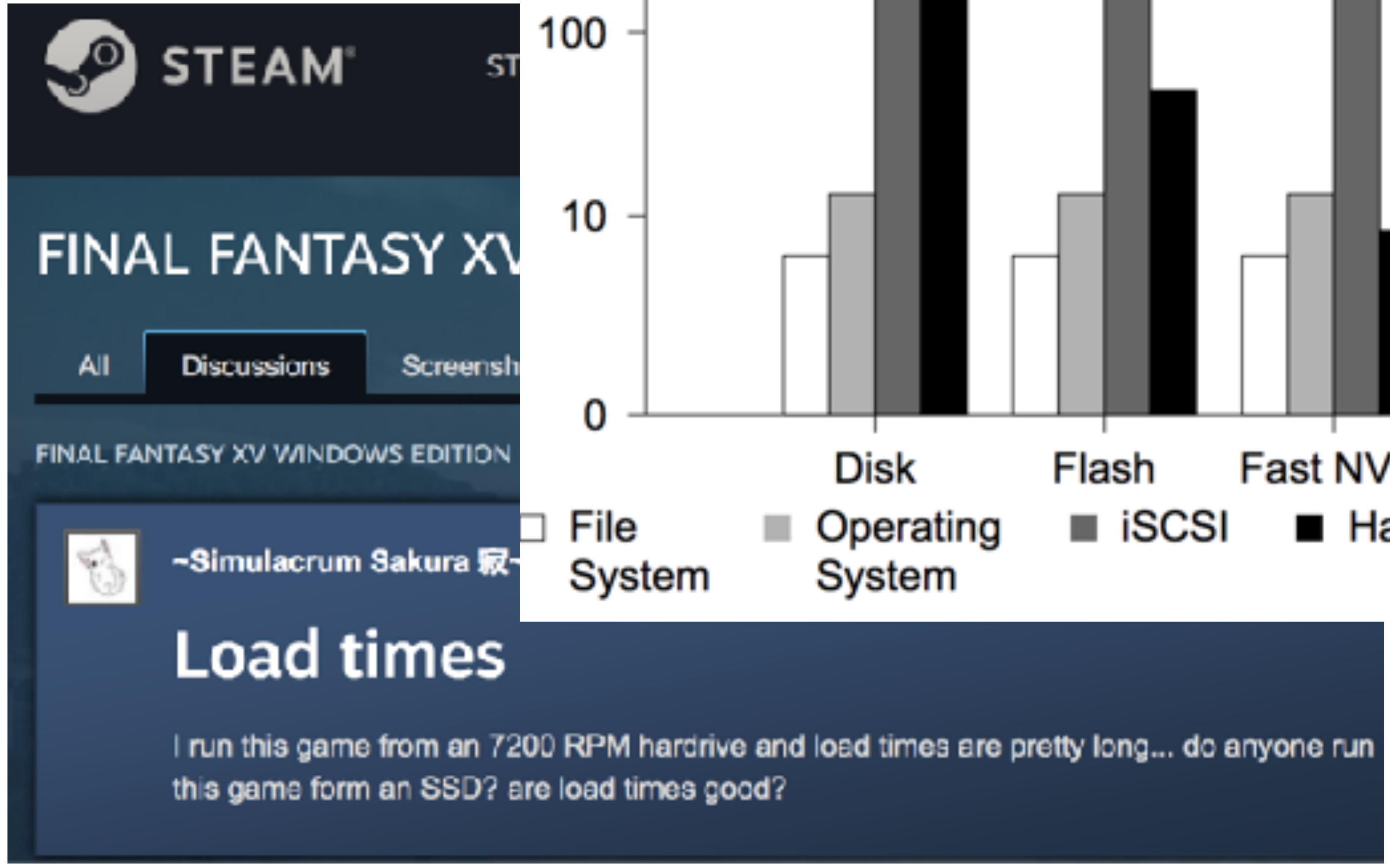
I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



https://www.pollev.com/hungweitseng close in 01:00

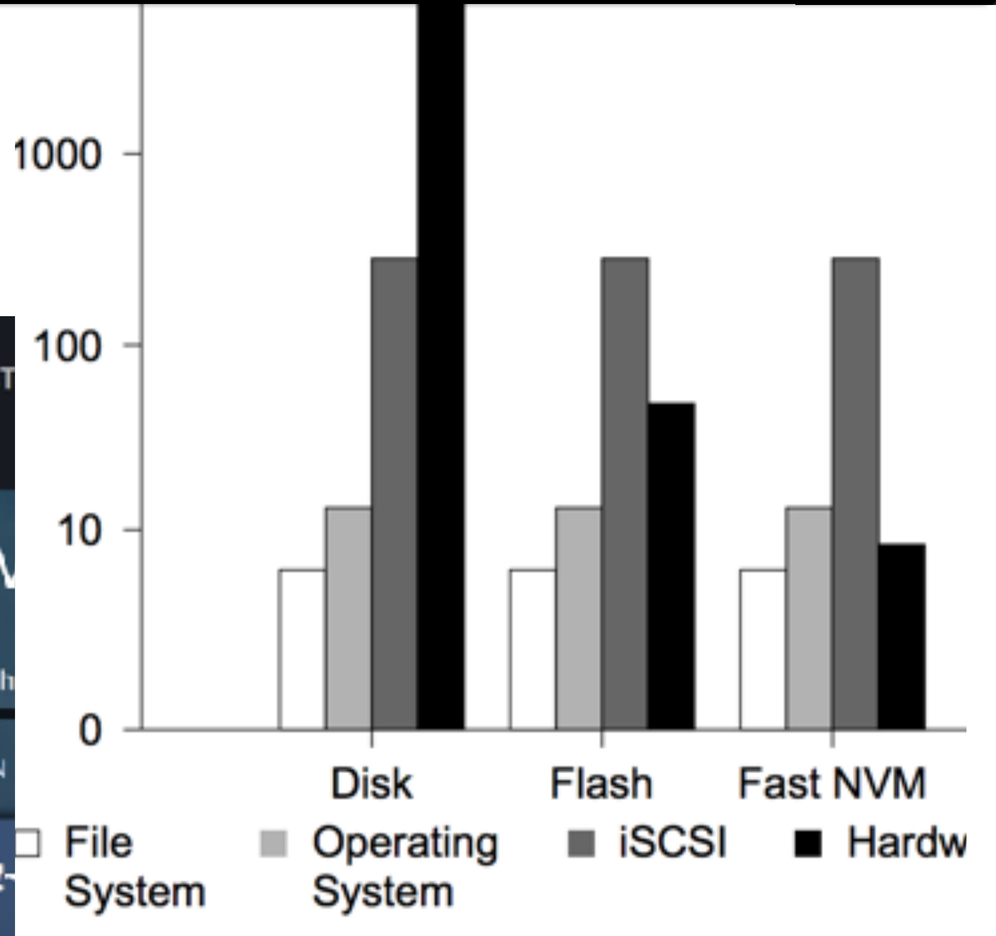
Speedup further!

- With the latest flash memory technologies, the system spends 16% of time on accessing the flash, and the software overhead is now 84%. If we want to adopt a new memory technology to replace flash to achieve 2x speedup on loading maps, how much faster the new technology needs to be?
  A. ~5x
  B. ~10x
  C. ~20x
  D. ~100x
  E. None of the above



Load times

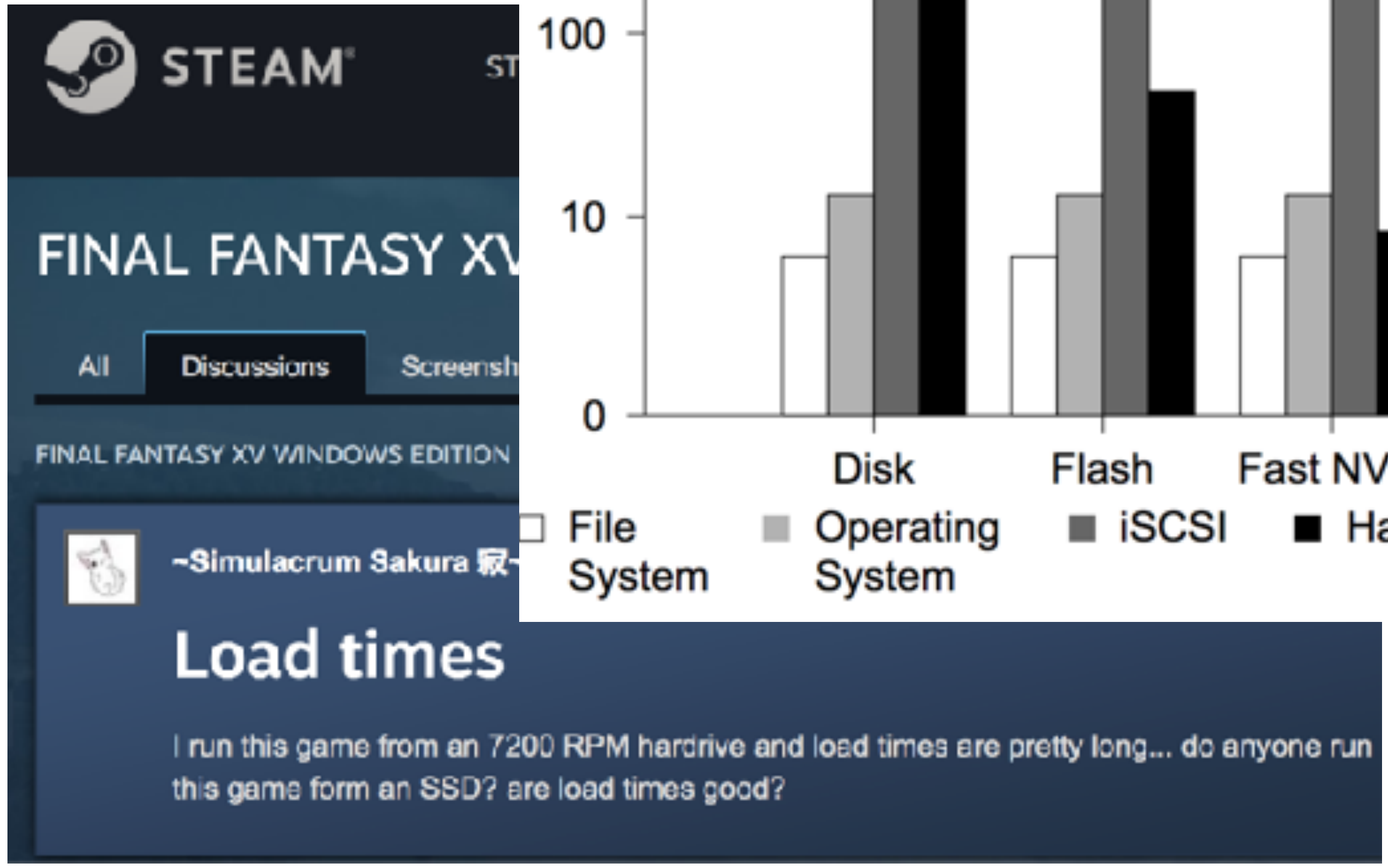
I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



https://www.pollev.com/hungweitseng close in 01:00

Speedup further!

- With the latest flash memory technologies, the system spends 16% of time on accessing the flash, and the software overhead is now 84%. If we want to adopt a new memory technology to replace flash to achieve 2x speedup on loading maps, how much faster the new technology needs to be?
  - A. ~5x
  - B. ~10x
  - C. ~20x
  - D. ~100x
  - E. None of the above


STEAM

FINAL FANTASY XV

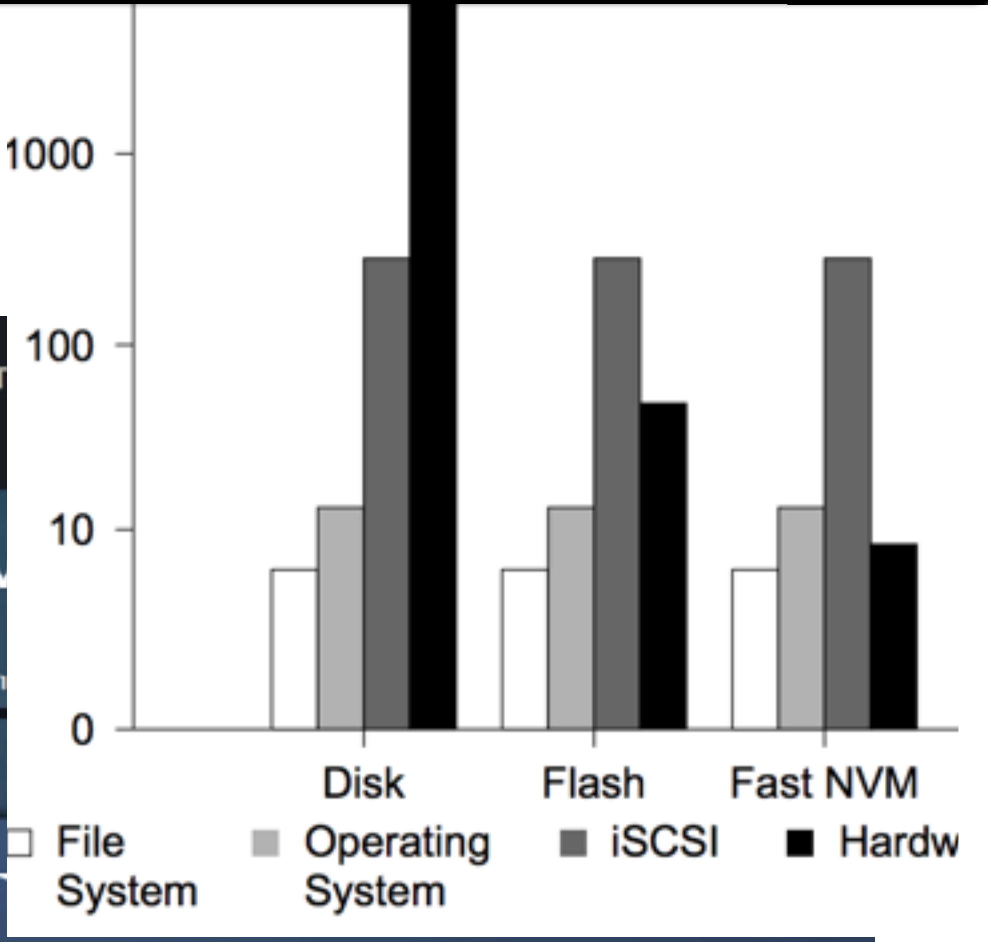
All | Discussions | Screensh

FINAL FANTASY XV WINDOWS EDITION

~Simulacrum Sakura 寂~

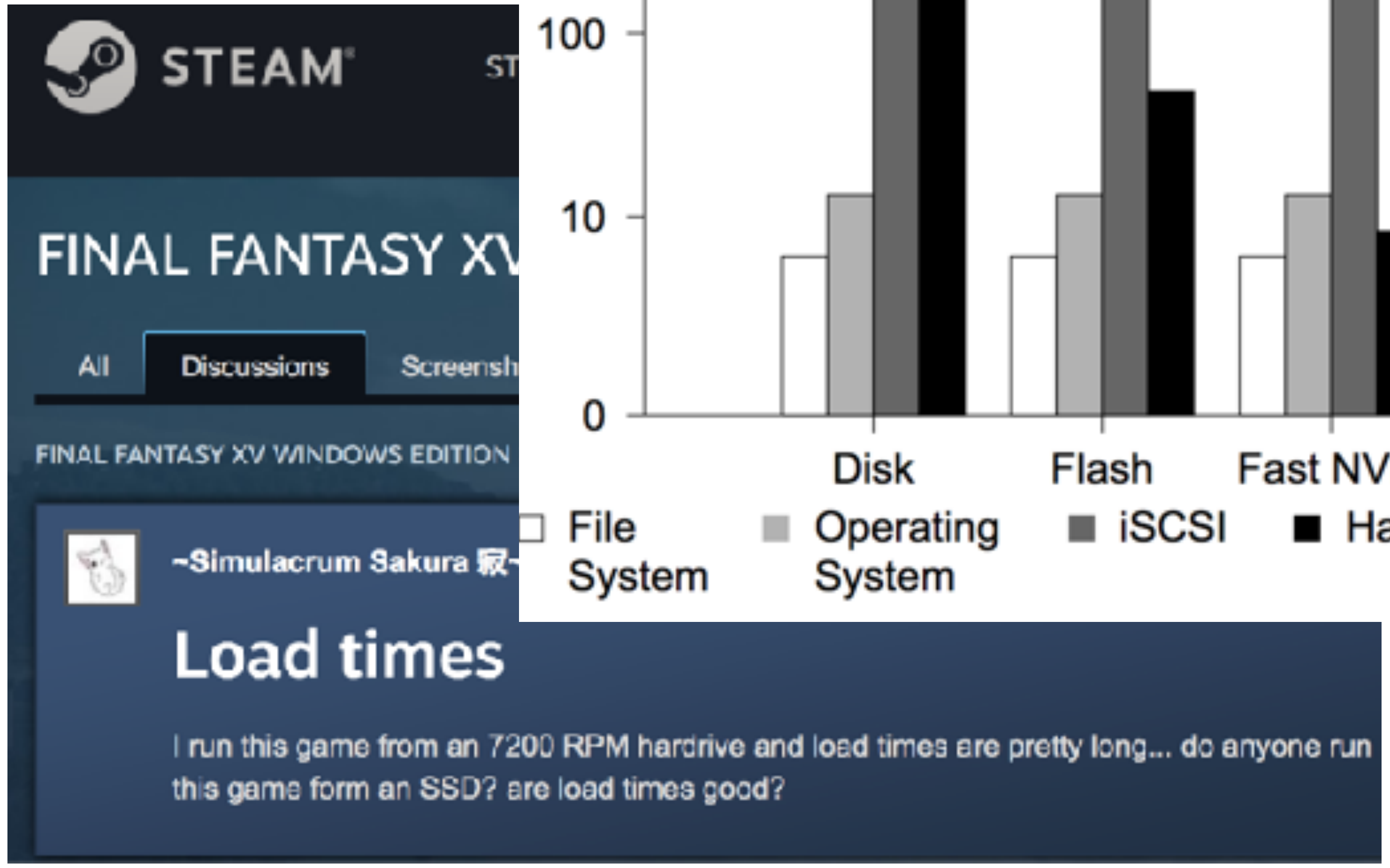
Load times

I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



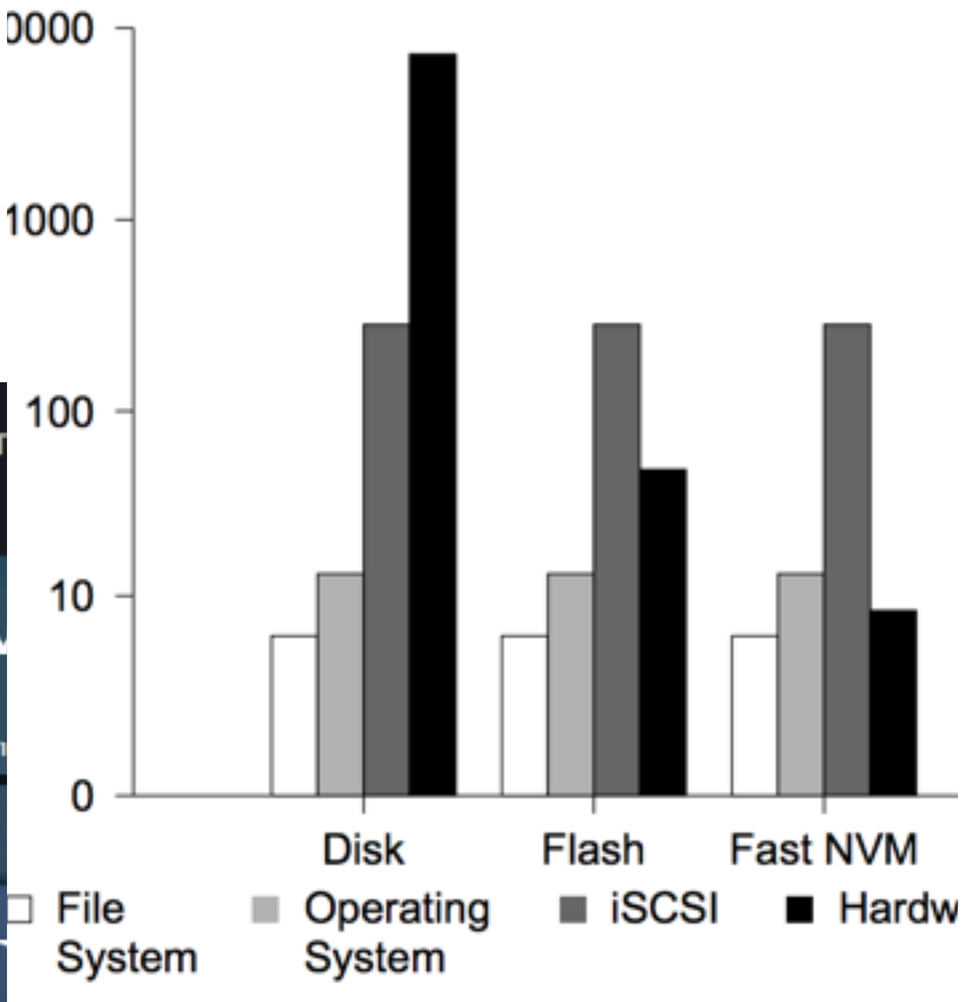
Speedup further!

- With the latest flash memory technologies, the system spends 16% of time on accessing the flash, and the software overhead is now 84%. If we want to adopt a new memory technology to replace flash to achieve 2x speedup on loading maps, how much faster the new technology needs to be?
  - A. ~5x
  - B. ~10x
  - C. ~20x
  - D. ~100x
  - E. [✓ None of the above]



Load times

I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?



$$Speedup_{enhanced}(16\%, x) = \frac{1}{(1-16\%)+\frac{16\%}{x}} = 2$$

$$x = 0.47$$

Does this make sense?

Amdahl's Law Corollary #1

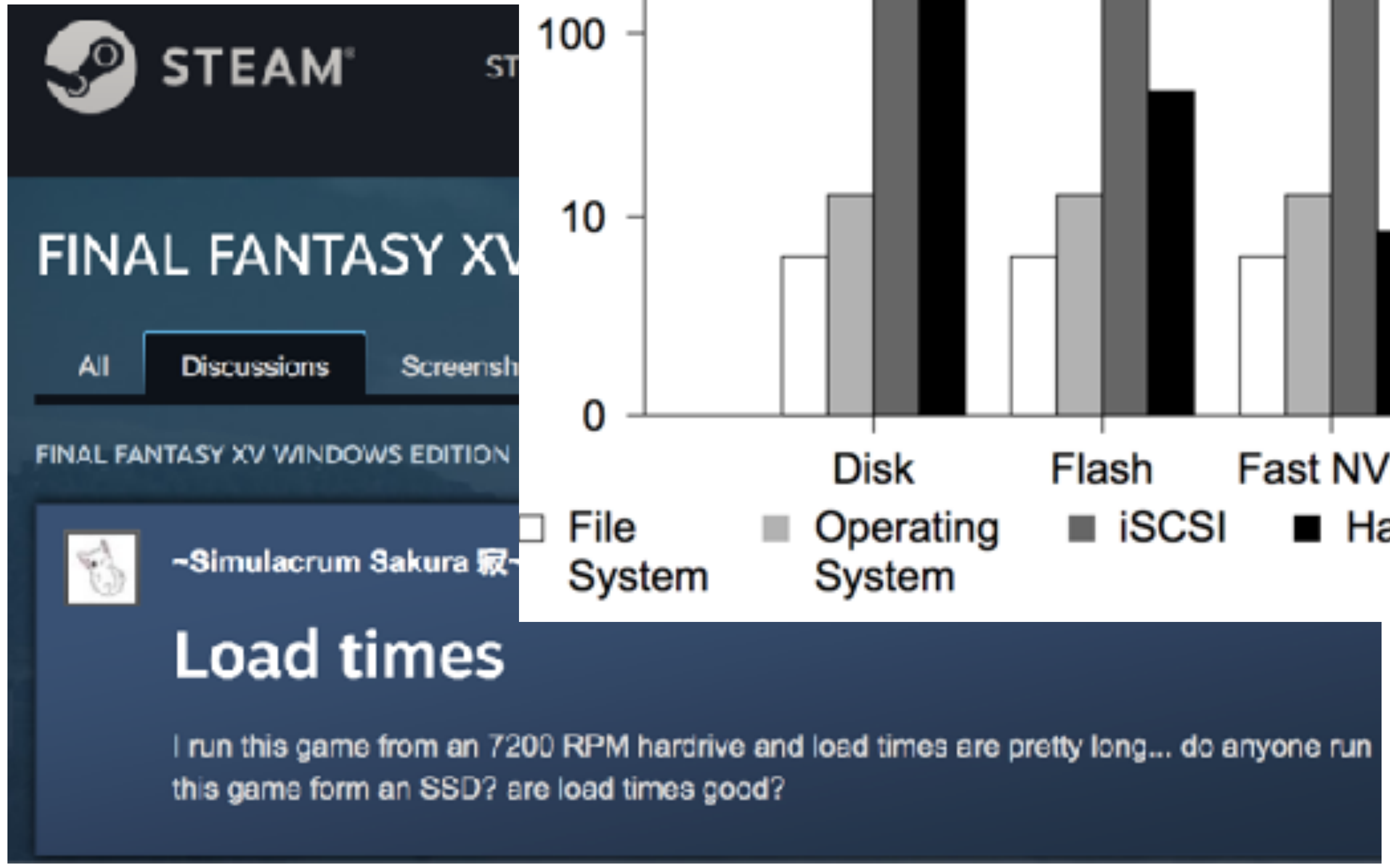
- The maximum speedup is bounded by

$$Speedup_{max}(f, \infty) = \frac{1}{(1-f) + \frac{f}{\infty}}$$

$$Speedup_{max}(f, \infty) = \frac{1}{(1-f)}$$

Speedup further!

- With the latest flash memory technologies, the system spends 16% of time on accessing the flash, and the software overhead is now 84%. If we want to adopt a new memory technology to replace flash to achieve 2x speedup on loading maps, how much faster the new technology needs to be?
  A. ~5x
  B. ~10x
  C. ~20x
  D. ~100x
  [✓ E. None of the above]


STEAM ST

FINAL FANTASY XV

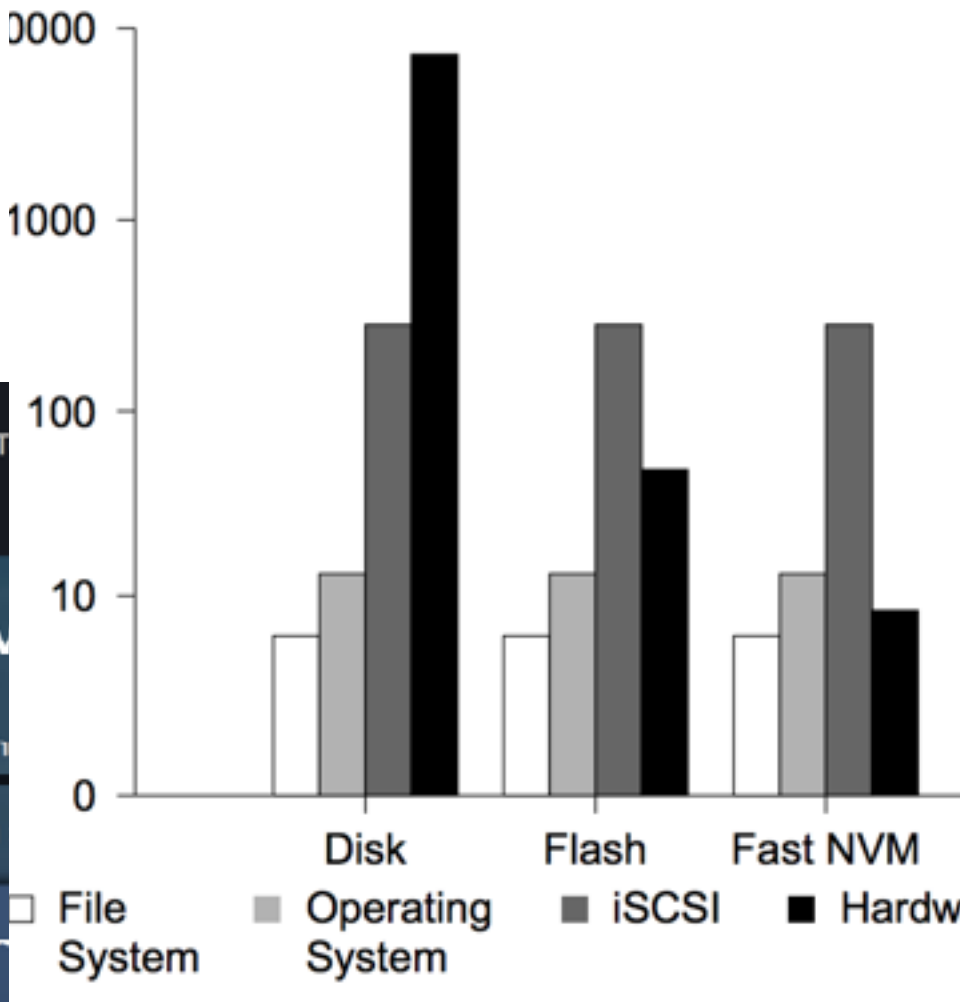
All Discussions Screensh

FINAL FANTASY XV WINDOWS EDITION

~Simulacrum Sakura 寂~

Load times

I run this game from an 7200 RPM hardrive and load times are pretty long... do anyone run this game form an SSD? are load times good?

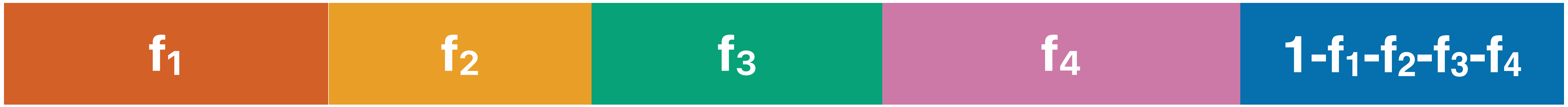


$$Speedup_{max}(16\%, \infty) = \frac{1}{(1-16\%)} = 1.19$$

2x is not possible

Corollary #1 on Multiple Optimizations

- If we can pick just one thing to work on/optimize



$$Speedup_{max}(f_1, \infty) = \frac{1}{(1 - f_1)}$$

$$Speedup_{max}(f_2, \infty) = \frac{1}{(1 - f_2)}$$

$$Speedup_{max}(f_3, \infty) = \frac{1}{(1 - f_3)}$$

$$Speedup_{max}(f_4, \infty) = \frac{1}{(1 - f_4)}$$

The biggest f_x would lead to the largest $Speedup_{max}$!

Corollary #2 — make the common case fast!

- When f is small, optimizations will have little effect.
- Common == **most time consuming** not necessarily the most frequent
- The uncommon case doesn’t make much difference
- The common case can change based on inputs, compiler options, optimizations you’ve applied, etc.