I/O and MapReduce

CS110 Discussion 14

Suting Chen

I/O

- Hierarchy design

- Types of I/O

I/O Hierarchy design

- CPU register

- Cache

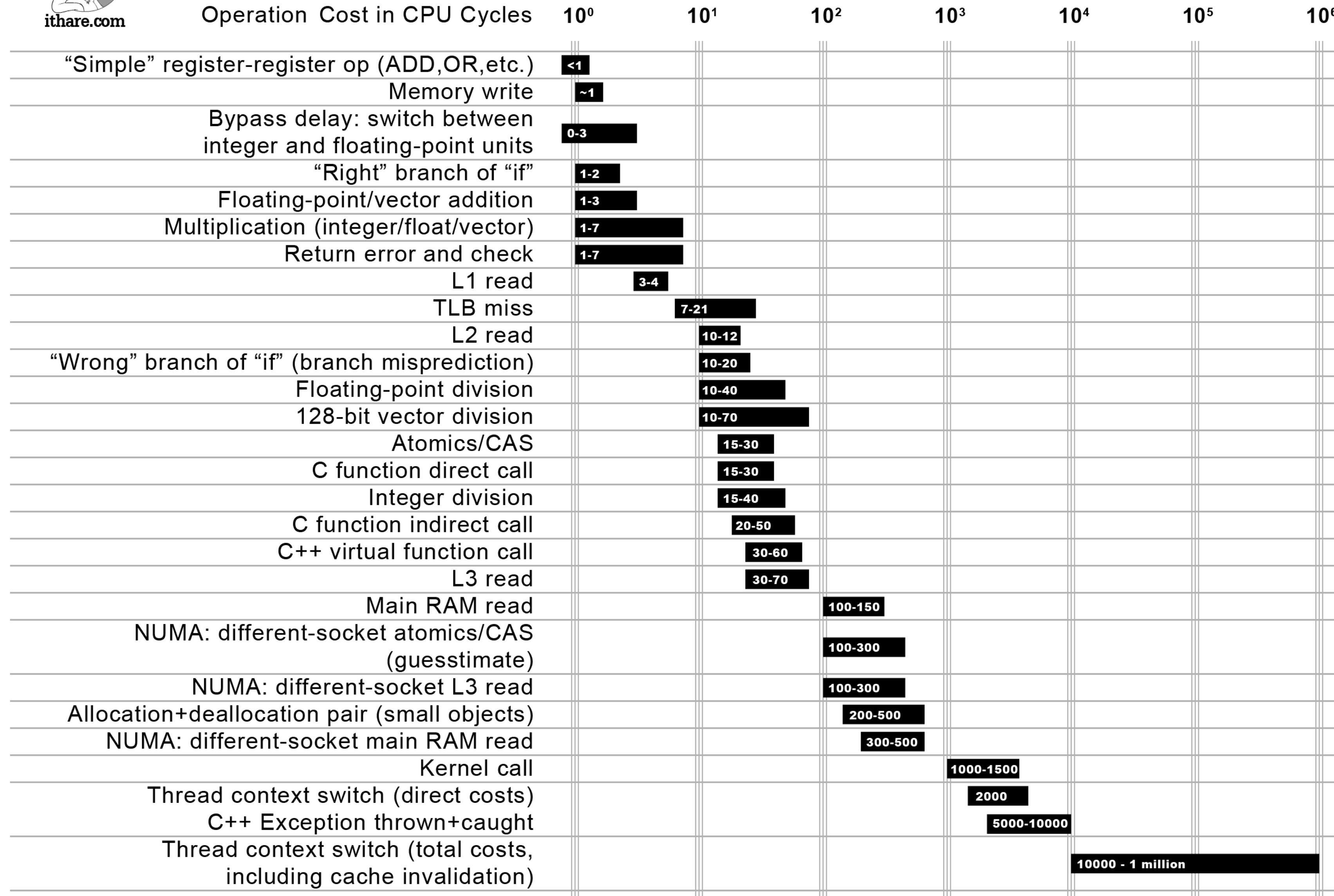- DRAM

- HDD/SSD

- CD/Tape

- Network storage

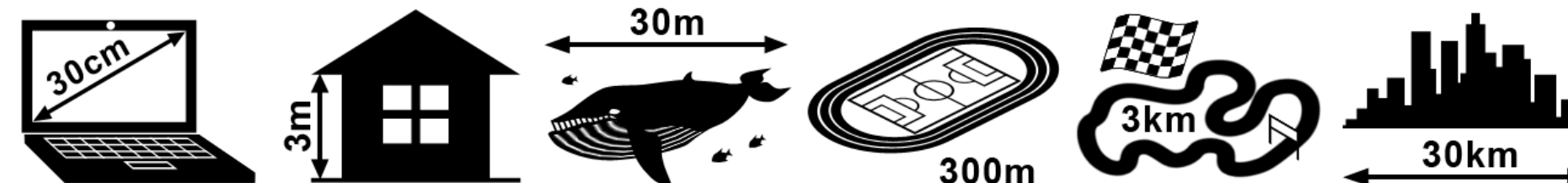Slower but larger

Not all CPU operations are created equal

ithare.com

Operation Cost in CPU Cycles	10^0	10^1	10^2	10^3	10^4	10^5	10^6
"Simple" register-register op (ADD,OR,etc.)	<1						
Memory write	~1						
Bypass delay: switch between integer and floating-point units	0-3						
"Right" branch of "if"	1-2						
Floating-point/vector addition	1-3						
Multiplication (integer/float/vector)	1-7						
Return error and check	1-7						
L1 read	3-4						
TLB miss	7-21						
L2 read		10-12					
"Wrong" branch of "if" (branch misprediction)		10-20					
Floating-point division		10-40					
128-bit vector division		10-70					
Atomics/CAS		15-30					
C function direct call		15-30					
Integer division		15-40					
C function indirect call		20-50					
C++ virtual function call		30-60					
L3 read		30-70					
Main RAM read			100-150				
NUMA: different-socket atomics/CAS (guesstimate)			100-300				
NUMA: different-socket L3 read			100-300				
Allocation+deallocation pair (small objects)			200-500				
NUMA: different-socket main RAM read			300-500				
Kernel call				1000-1500			
Thread context switch (direct costs)				2000			
C++ Exception thrown+caught				5000-10000			
Thread context switch (total costs, including cache invalidation)					10000 - 1 million		

Distance which light travels while the operation is performed

30cm   3m   30m   300m   3km   30km

Link

I/O
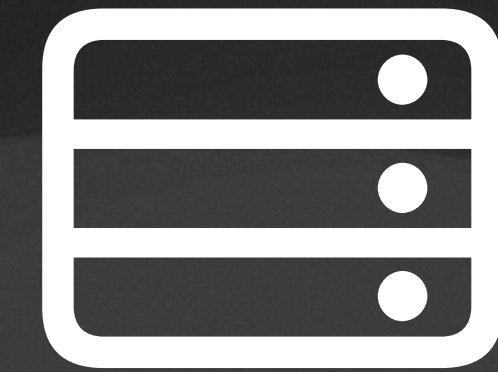
- Hierarchy design
- Types of I/O

Types of I/O

- Blocking Synchronous

- Non-blocking Synchronous

- Blocking Asynchronous

- Non-blocking Asynchronous

Types of I/O

Client

Server

Blocking / Non-blocking

- Focus on the behavior of client

- Blocking: Client must wait before any response is made.

- Non-blocking: Client can do anything else.

Synchronous / Asynchronous

- Focus on the server

- Synchronous: The server do the task immediately.

- Asynchronous: The server would "call back" when it's ready.

More details…

- Blocking and Synchronous

- Blocking and Asynchronous

- Non-blocking and Synchronous

- Non-blocking and Asynchronous

More details...
Blocking and Synchronous

- All the file operations you've ever done in C/C++.
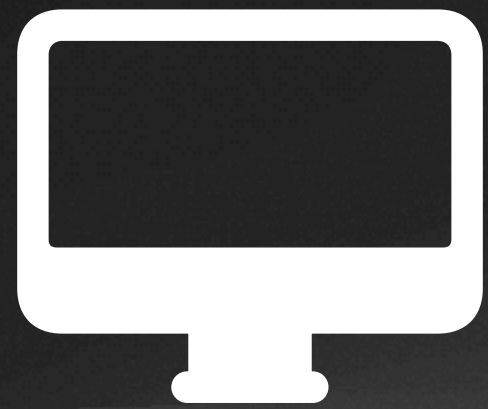
Client
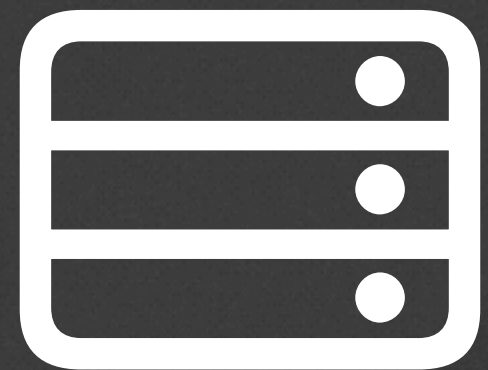
Server
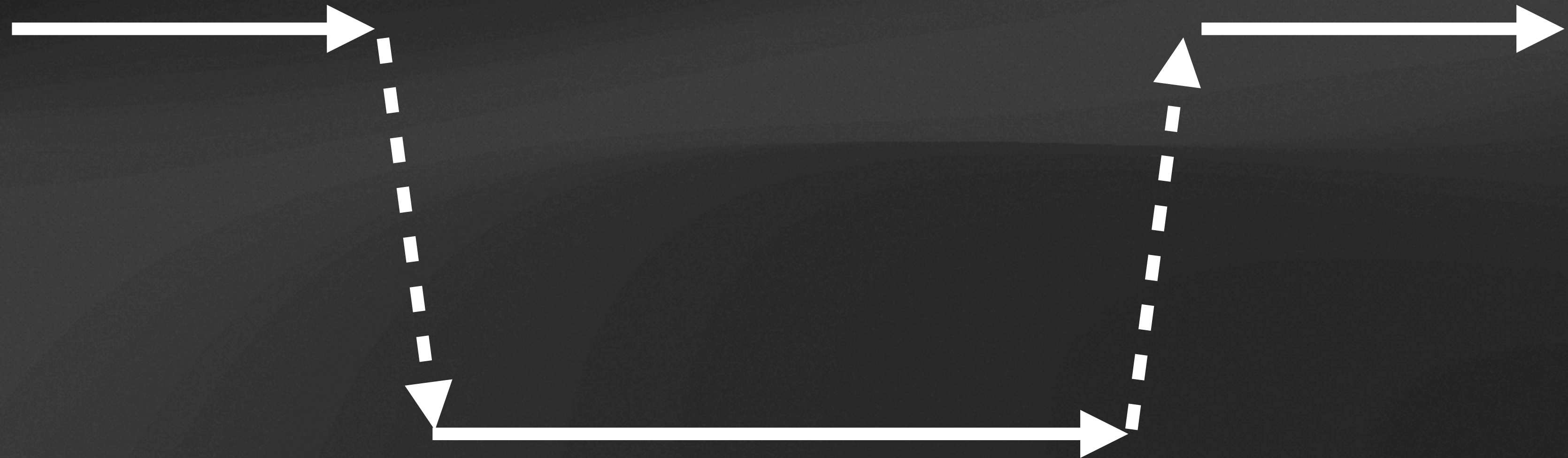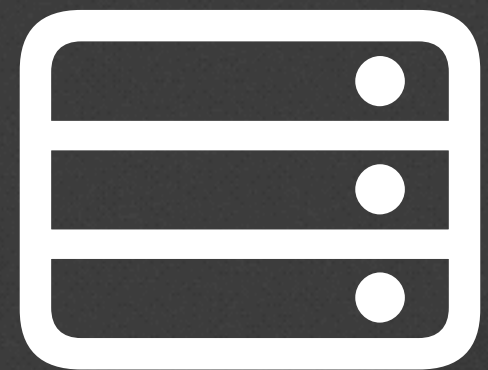
More details...
Blocking and Asynchronous

- I/O multiplexing

Client

Server

Readable

Actual Data
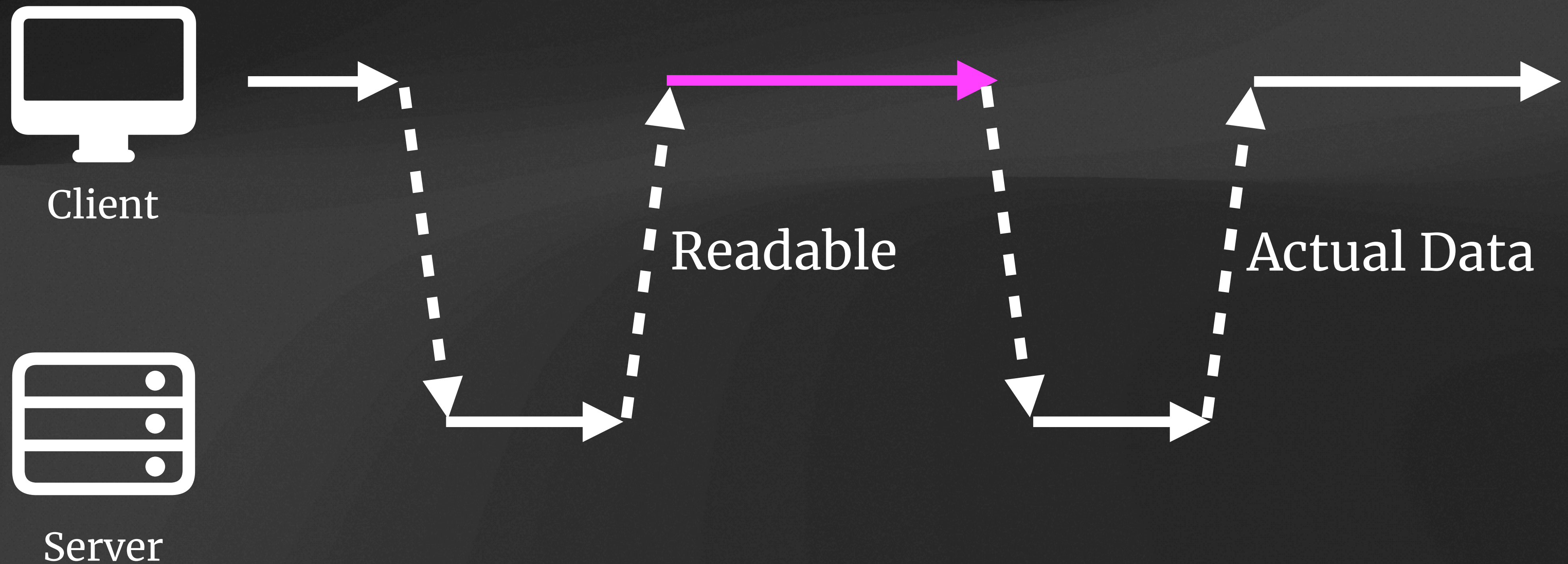
More details...
Non-Blocking and Synchronous
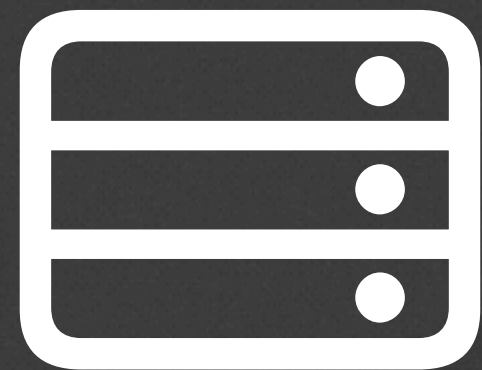
Client

Server
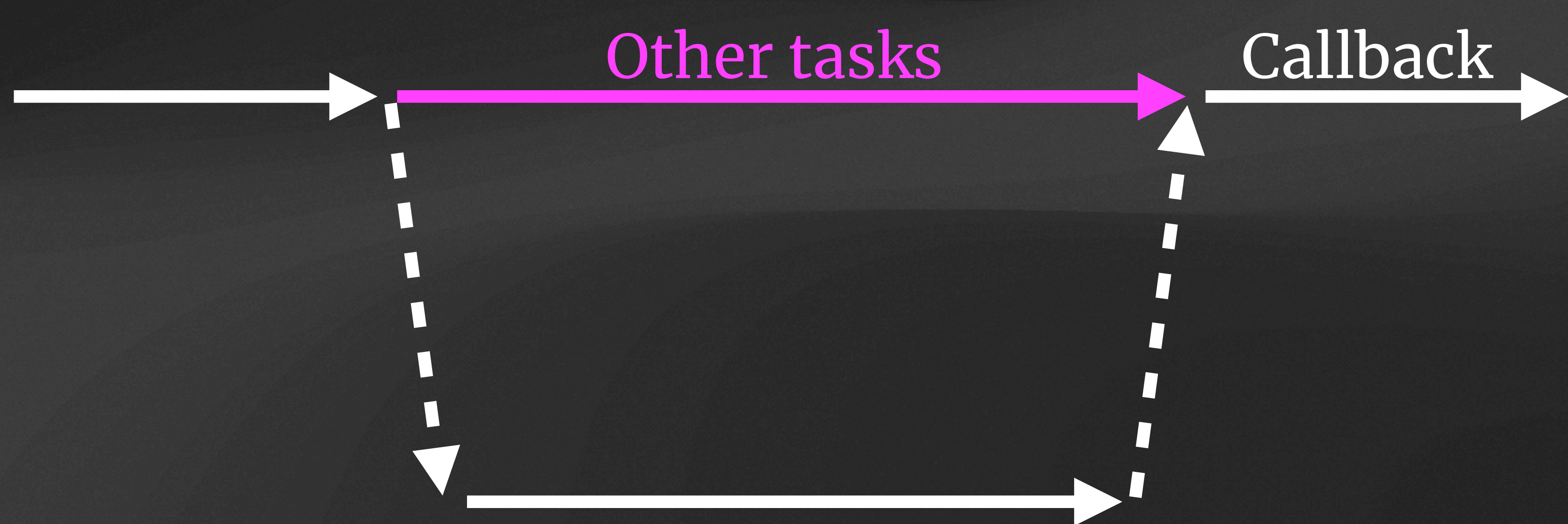
No result

No result

Result return

More details...
Non-Blocking and Asynchronous

MapReduce

- Map: Preprocess and split the task

- Reduce: Solve each sub-problem

MapReduce
Terasort

Node 1

Node 2

Node 3

34 48 64 97 82 29
65 52 91 21 8 31 88
29 12 66 57 95

MapReduce

Terasort

34 48
82 21
31 88

Node 1

64 29
91 52
8 67

Node 2

97 41
29 12
57 95

Node 3

MapReduce
Terasort

| Node 1 | 34 48
82 21
31 88 |

| Node 2 | 64 29
91 52
8 67 |

| Node 3 | 97 41
29 12
57 95 |

3 Nodes, so split into 3 groups:

1. 0~33

2. 34~66

3. 67~100

MapReduce
Terasort

Node 1

Node 2

Node 3

8 29
29 21
31 12

34 48
64 52
41 57

97 82
88 91
67 95

http://sortbenchmark.org/

MapReduce
Terasort

Node 1

Node 2

Node 3

8 12 21 29 29 31

34 41 48 52 57 64

67 82 88 91 95 97

http://sortbenchmark.org/

MapReduce
Terasort

8 12 21 29 29 31 34 41 48 52 57 64 67 82 88 91 95 97

" The best way to predict the future is to invent it. "

— Alan Curtis Kay, Computer Scientist, ACM Turing Award (2003)