

Programação para Cluster e Multicore

Programação para Cluster e Multicore e GPUs

Programação para Cluster e Multicore e GPUs e FPGAs

Programação para

~~Cluster~~

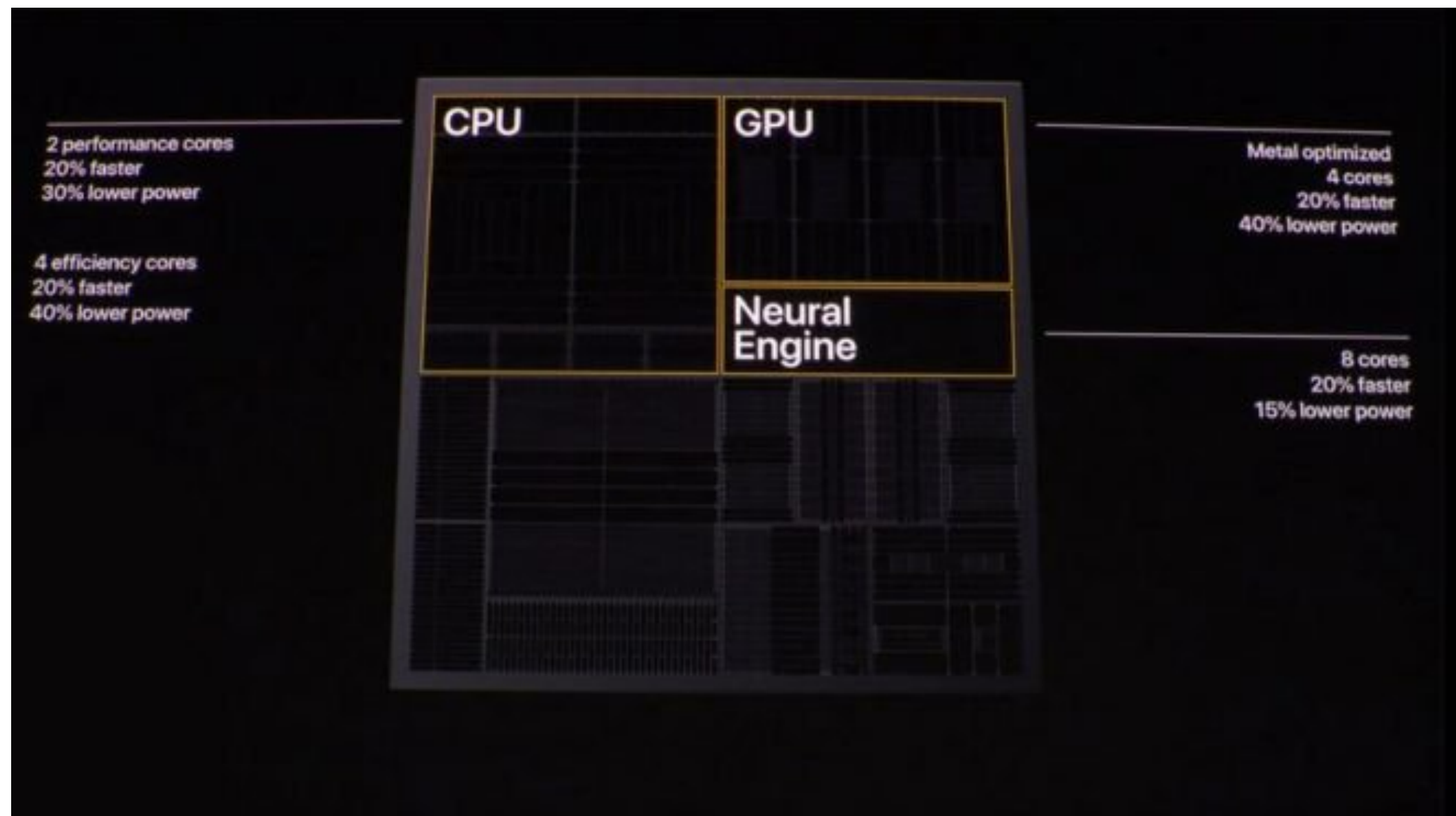
~~e Multicore~~

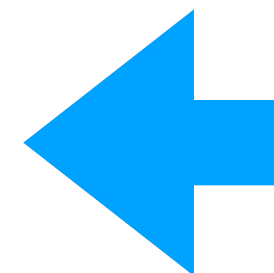
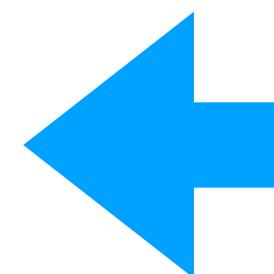
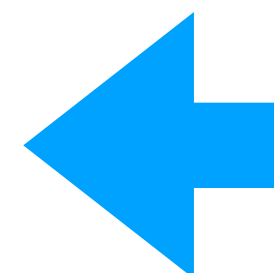
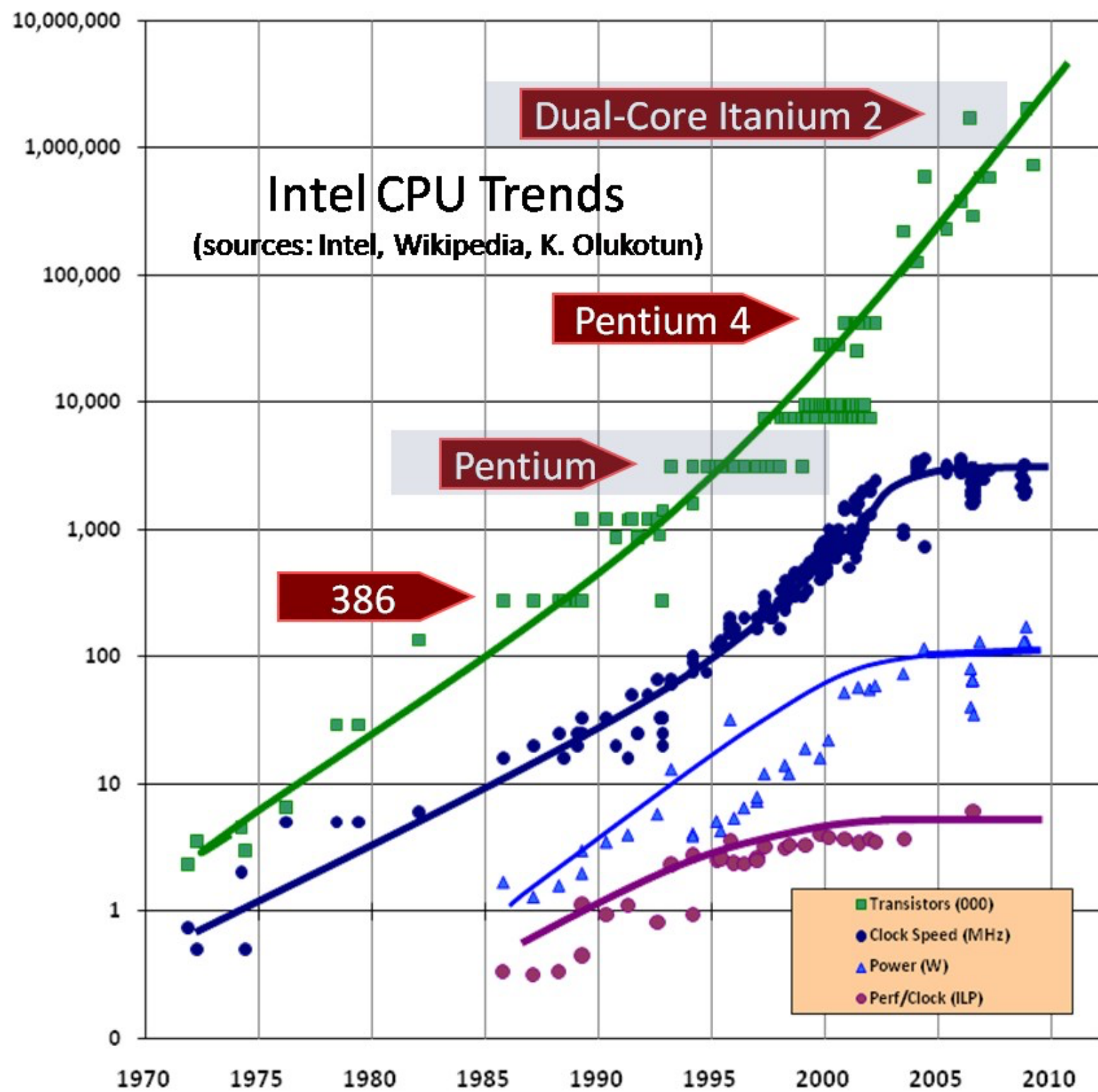
~~e GPUs~~

~~e FPGAs~~

Paralela e Concorrente

Apple A13 Bionic (iPhone 11)





What do these 2017 smartphones have in common?



**Xiaomi
Redmi Note 4**



Meizu Pro 6



Elephone S7

10 core CPUs



**Xiaomi
Redmi Note 4**



Meizu Pro 6



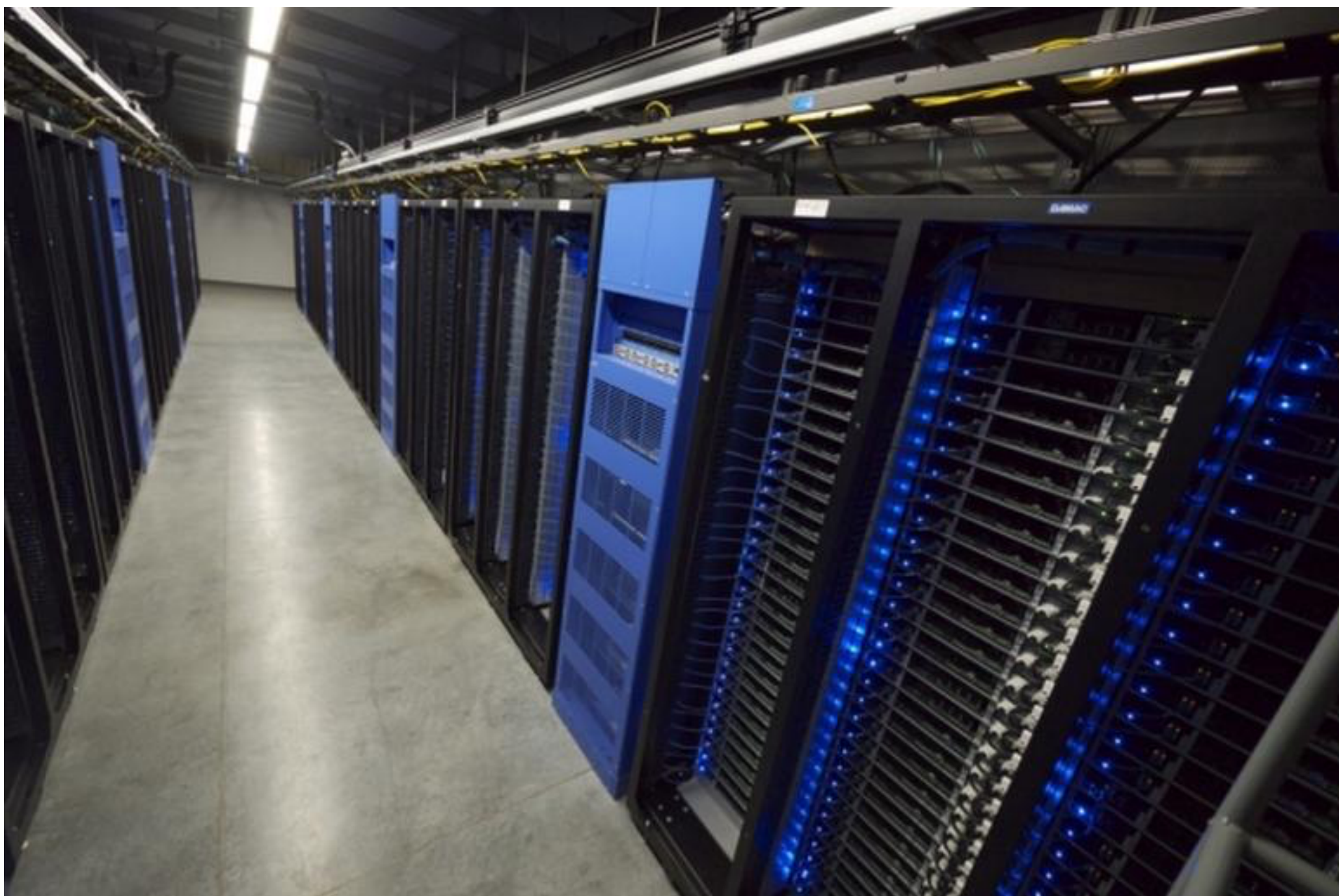
Elephone S7

Hey Siri!
Alexa,
OK Google,



Siri Not Available

You are not connected to the Internet.





Google

2018 BEST PLACES TO WORK

Engaged Employer

Overview

9.4k
Reviews

11k
Jobs

19k
Salaries

8.9k
Interviews

2.9k
Benefits

330
Photos

Follow

+ Add an Interview

Interview Question

Engineering Manager/Director, Site Reliability Engineering Interview Dublin, Co. Dublin (Ireland)

"How many computers would it take to run gmail."

Answer | Add Tags



Work in HR or Marketing?

Get a [free employer account](#)



How many computers would it take to run Gmail?

Assume that gmail has 2 bln of users (there are 3 bln internet users around the world and almost every internet user around the world has a gmail account). Assume that average size of mailbox is 500Mb (I've worked at many email services and know what I'm saying). 2 bln by 500Mb is 1000pb (petabytes).

If gmail stores each email 3 times (2 times is a good choice, but since users' mailboxes are scattered around the storage cluster then 2 collapsed hard disks at the same moment could result in some data lost while the possibility of 3 hard disk have collapsed in the different data centers at the same time is negligibly low) then total is 3000pb. But I'm completely sure that gmail has deduplication and gmail has text parts of emails zipped. The first measure can save around 50% of space (the most of the mail store - it's attachments, and attachments are usually sent by person, so an attachment appears twice - in an inbox folder and in a sent folder). The second measure can save around 20% of space (this is the approximate total size of all text parts of mailings that could loose their weight by times). So 3000Pb are reduced by 70% to again 1000Pb. A 36x4Tb storage server has a 144Tb capacity. 1000Pb divide by 144Tb is about 7000. Why is it 36x4Tb? Because 36 is as far as I know economically feasible maximum number of disks per server. And 4Tb is economically feasible maximum disk size. I mean here that the more disks per server the less is the cost of the server other-than-disk parts per byte of storage but to the limit after that other-than-disk parts become more expensive. So is about the disk size - the bigger the disk size the cheaper is the byte stored on it, but to the limit of 4Tb.

Of course gmail is not only storage service. There are also web services, smtp, imap, pop3, MX, antispam and so on. All of these services just process data and they don't have to store anything large (maybe there are some databases negligible compared to the email storage to store user credentials for example). A service does not necessarily needs a separate server. And storage servers luckily have a plenty of CPU time and some memory. So, the storage server's CPU time, memory and small portion of the disk could be used for data processing and for some small databases needed to web services, smtp, pop3, imap and so on. Google would not be Google, if they used a lot of additional servers for the life of the email system, except the email storage servers.

That's why the correct answer is **7000** servers.

How to write scalable software?

- **Parallelism**

- Two operations **occurring** at the same time
- Multicore CPUs, GPUs, FPGAs, clusters
- OpenMP, SIMD, OpenCL, Cuda
- Usually about performance

- **Concurrency**

- Two interleaving operations can be **in progress** at the same time
- Any hardware
- Multiprocess, threads, actors
- Usually about correctness and modularity

1	[0.7%]	7	[0.0%]	13	[0.0%]	19	[0.0%]
2	[1.3%]	8	[0.0%]	14	[0.0%]	20	[0.0%]
3	[0.0%]	9	[0.0%]	15	[1.3%]	21	[0.0%]
4	[0.0%]	10	[0.0%]	16	[0.0%]	22	[0.0%]
5	[0.0%]	11	[0.0%]	17	[0.0%]	23	[0.0%]
6	[0.0%]	12	[0.0%]	18	[0.0%]	24	[0.0%]

Franklin is a 12 core machine with hyper-threading

```
for(int i=0; i<24; i++) {
    Runnable runnable = () -> {
        TimeUnit.SECONDS.sleep(1);
        System.out.println("Thread " + i);
    };
    Thread thread = new Thread(runnable);
    thread.start();
}
```

This is parallelism

```
for(int i=0; i<25; i++) {
    Runnable runnable = () -> {
        TimeUnit.SECONDS.sleep(1);
        System.out.println("Thread " + i);
    };
    Thread thread = new Thread(runnable);
    thread.start();
}
```

This is concurrency

Programação para

~~Cluster~~

~~e Multicore~~

~~e FPGAs~~

Paralela e Concorrente

Preliminary Plan

- Cooperative Multithreading
- Divide and Conquer parallel algorithms
- Parallel Reduction
- Locking
- Java's synchronized and volatile
- ForkJoin, Futures and Dataflow
- Worksharing & Workstealing
- Producer/Consumer queues and channels
- Communicating Sequential Processes
- Actor Model
- Map-Reduce
- Multi-threaded Non-blocking IO
- MPI
- GPGPU Programming
- FPGA Programming
- Concurrent Program Debugging and Verification

Grading

$$\text{Final Grade} = (0.6 * \text{Paper} * \text{Presentation}) + (0.4 * \text{Assignments})$$

- Maximum of 8 pages
- Pre-approved topic
- 2 authors
- deadline two days before exam

- about the paper
- with discussion
- individual grading
- on the exam date

- 4 assignments
- individual
- done in TP and at home

Alcides Fonseca



Contactos

Departamento de Informática

Sala/Gabinete 6.3.17

Ext. Principal 26702

Email alcides@ciencias.ulisboa.pt

[Página Pessoal](#)

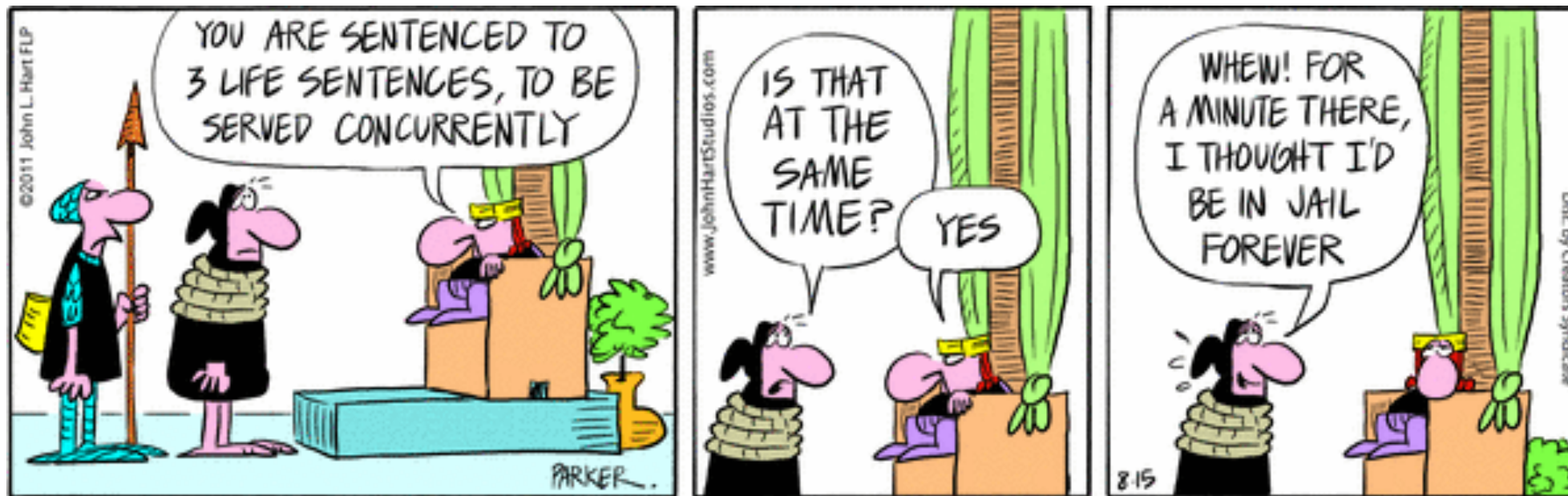
Office Hours:

Wednesdays, 10h-12h, Room C6.3.17

PCM-related mail:

docentes-pcm@listas.di.ciencias.ulisboa.pt

Until next week...



–Wizard of Id