Working with data at a low level

How computer helps you to deal with data stuff

Nikolay Markov, 2020

(pic source)

Shameless plug

- My name is **Nikolay Markov** and my nick is **@enchantner** almost everywhere on the Internet
- I work as a Principal Architect at **Aligned Research Group**
- For more than 9 years I'm using Python, also Golang, C/C++, Rust and Scala
- Dealing with clouds, distributed computing and networking (AWS, Azure, Kubernetes)
- Writing articles, reading lectures, organizing events
- Like chatting about system design, architecture and other low-level magic

(pic source)

Small: never use for, iterrows(), itertuples() и apply()

for loop: 645 ms ± 31 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

iterrows(): 166 ms ± 2.42 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

itertuples(): https://medium.com/@formigone/stop-using-df-iterrows-2fbc2931b60e

apply(): 90.6 ms ± 7.55 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)

https://bit.ly/2v4ZvLQ

(pic source)

SIMD & BLAS

What if we will apply one operation to multiple values?

MMX/SSE/AVX...

NeCPP - BLAS discussion

cuBLAS

Numpy + BLAS

Numpy + Intel MKL (+Install guide)

Multiplying matrices

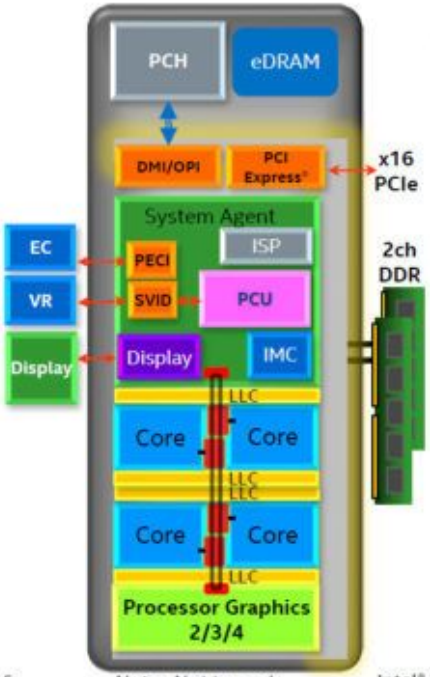
OpenBLAS

(pic source)

Intel and its magic

[Awesome article on AVX (in russian)]

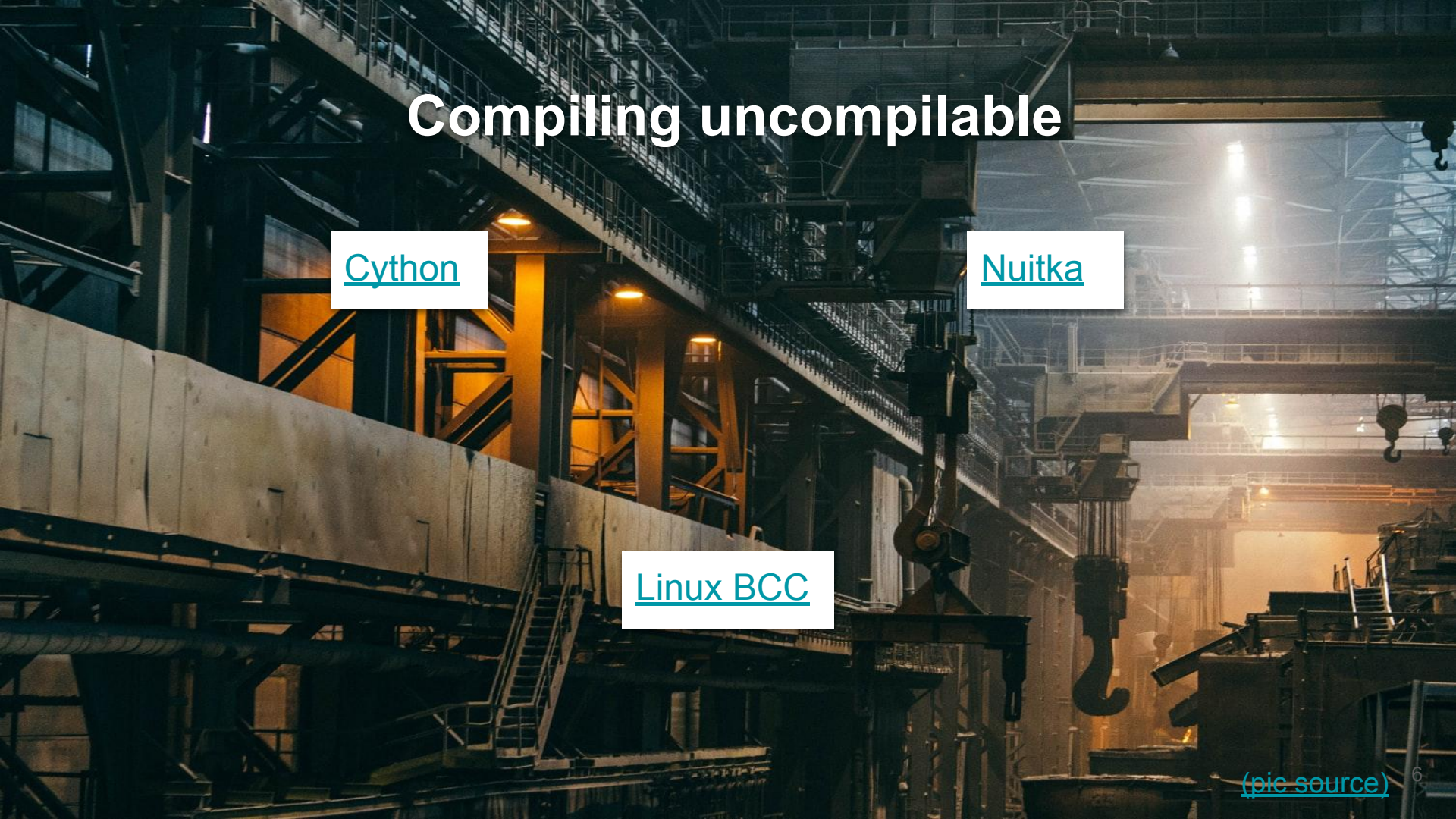
Skylake Overview – Power Management View



- Skylake is a SoC consisting of:
  - 2-4 CPU cores, Graphics, media, Ring interconnect , cache
  - Integrated System Agent (SA)
  - On package PCH and eDRAM
- Improved performance with aggressive power savings
- Package Control Unit (PCU) :
  - Power management logic and controller firmware
  - Continues tracking of internal statistics
  - Collects internal and external power telemetry: iMon, Psys
  - Interface to higher power management hierarchies: OS, BIOS, EC, graphics driver, DPTF, etc.

Note: Not to scale

Compiling uncompilable



Cython

Nuitka

Linux BCC

(pic source)

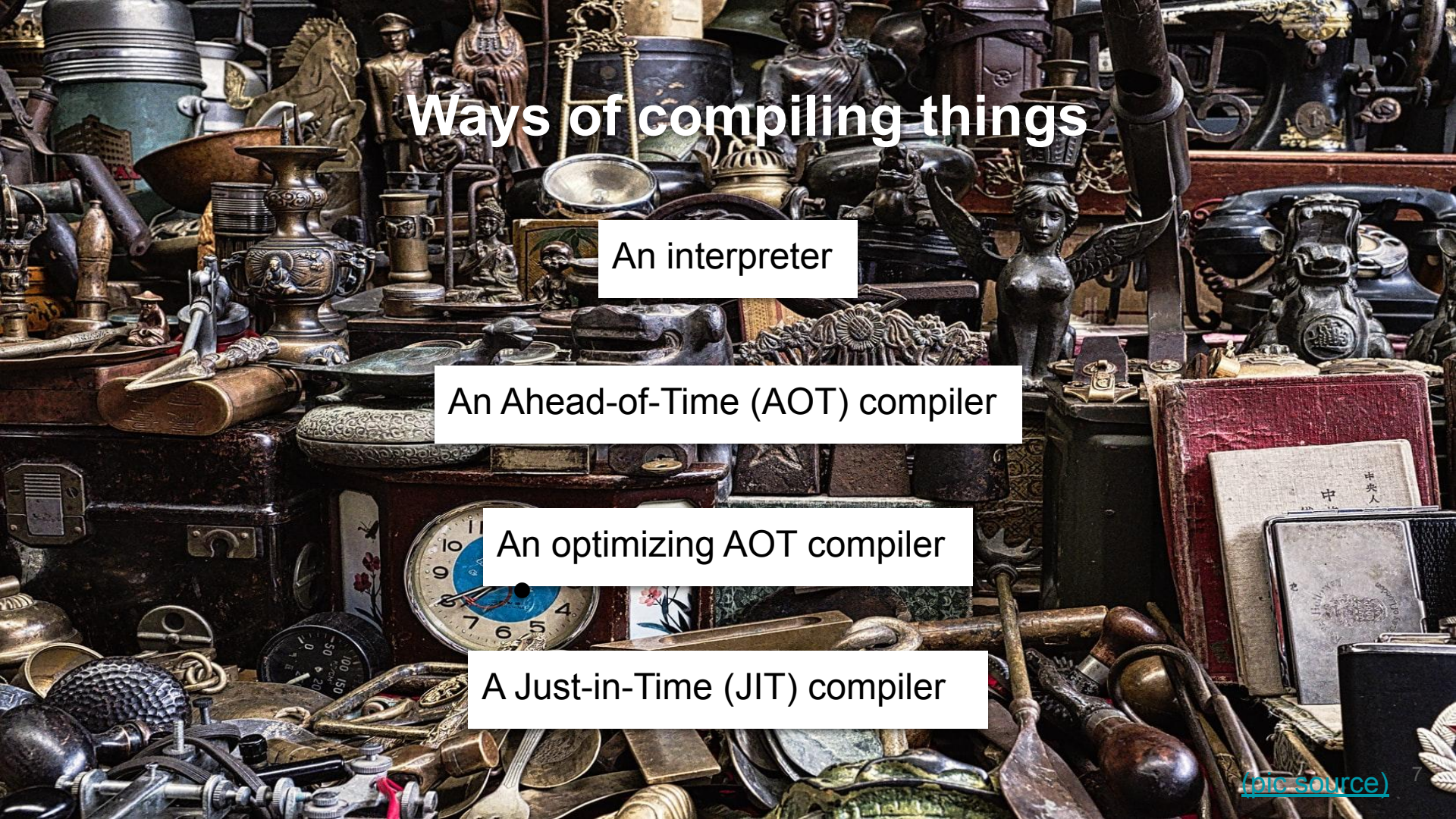
Ways of compiling things

An interpreter

An Ahead-of-Time (AOT) compiler

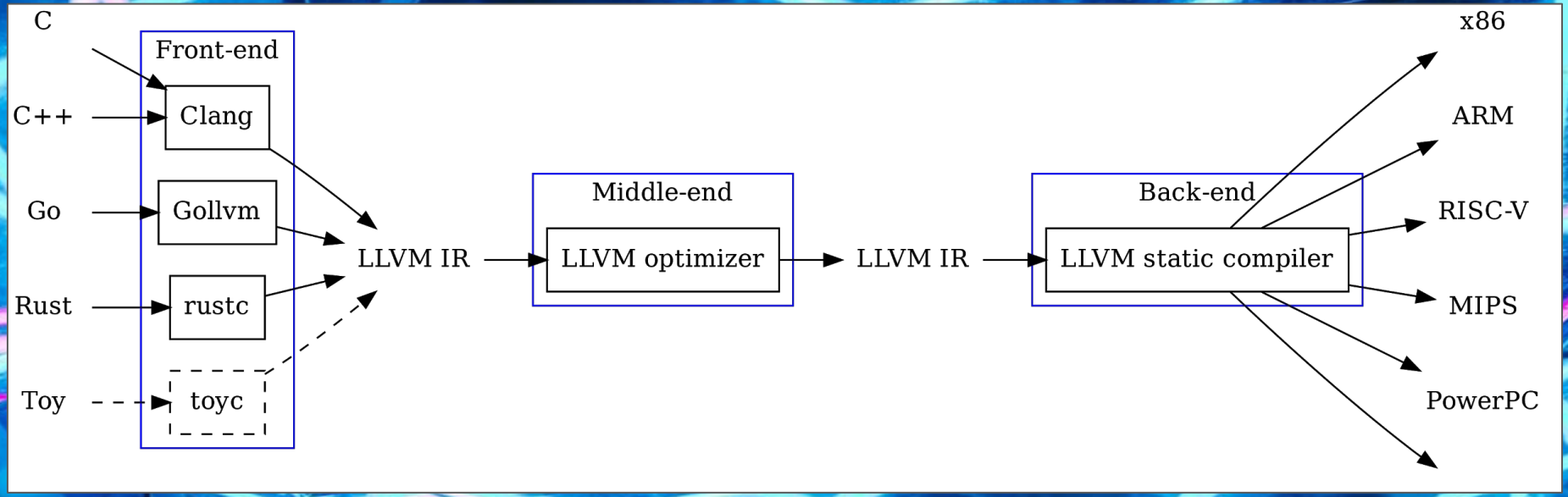
An optimizing AOT compiler

A Just-in-Time (JIT) compiler



(pic source)

Gimme a normal compiler, please



LLVM & Go

(pic source)

Just-in-time compilation

Tracing JIT + PyPy Paper

.NET - RyuJIT

LuaJIT (old, but gold)

Python world: PyPy (RPython), also Numba

News from Java - GraalVM (+explanation)

(pic source)

Interesting stuff to read - DMA

Using CPU to access memory is expensive

Imagine if we would have to do it for every network packet on a highly loaded server

Wouldn’t it be easier to read/write network card directly?

Use userspace TCP stack if you want to get more crazy

(pic source)

Interesting stuff to read - SMP

All processors are equal and homogenous

All CPUs have access to the same memory

(pic source)

Interesting stuff to read - NUMA

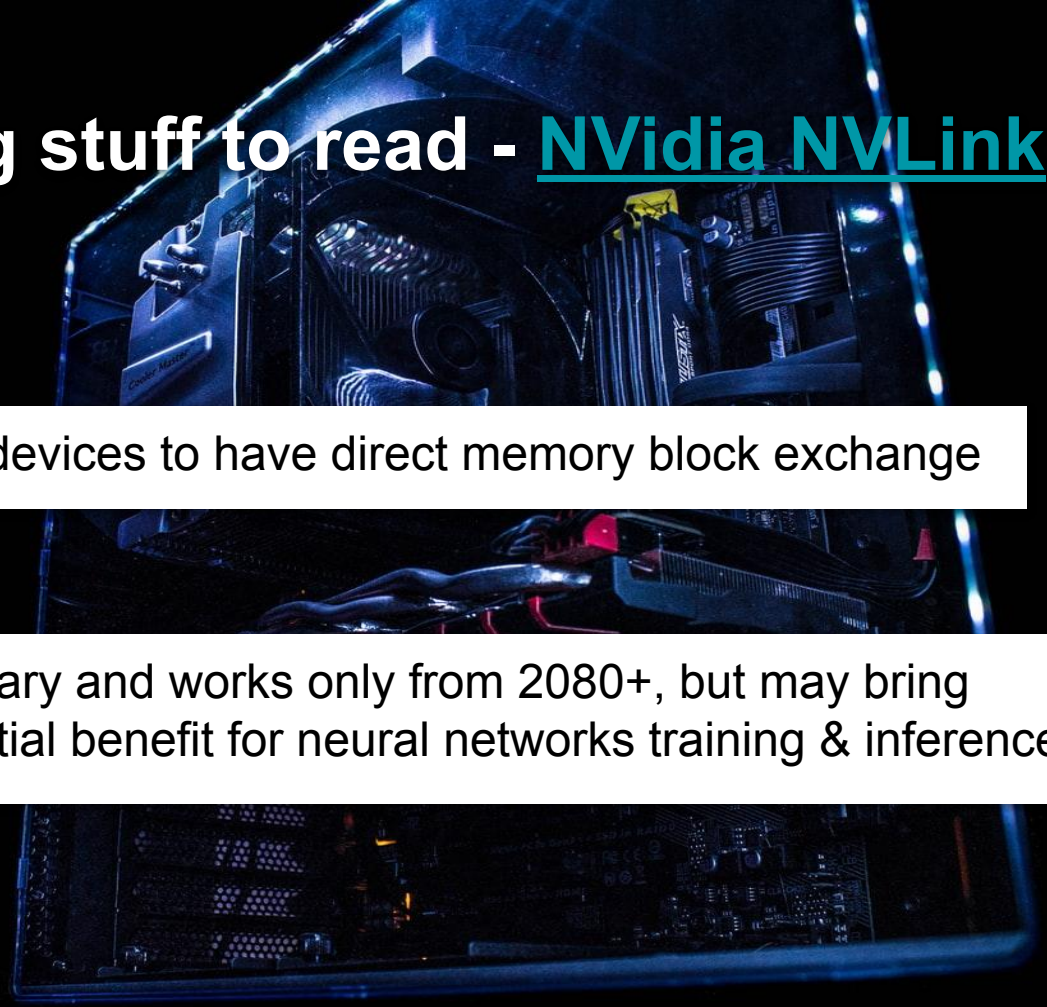
Every processor has its own memory

Great performance benefits if used right

All problems of a distibuted infrastructure (cache coherency, etc.) on one machine

(pic source)

Interesting stuff to read - NVidia NVLink



Let multiple devices to have direct memory block exchange

Proprietary and works only from 2080+, but may bring substantial benefit for neural networks training & inference

(pic source)

Interesting stuff to read - USL

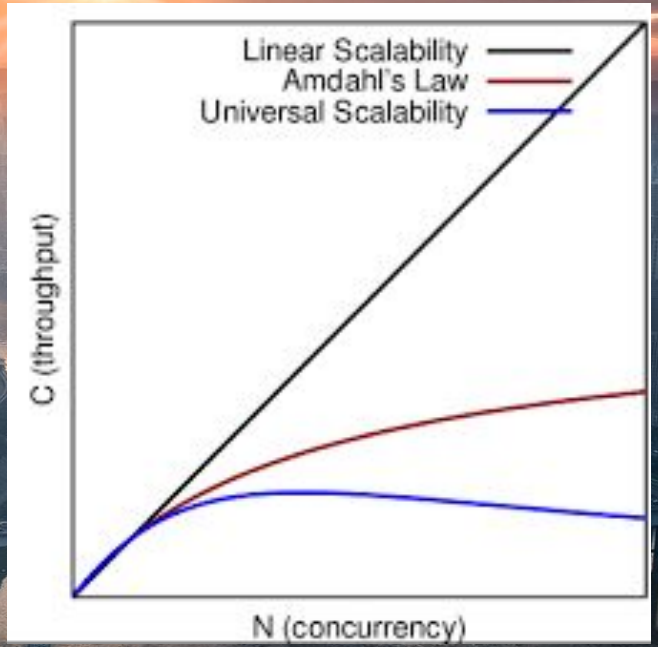
More common case for Amdahl's Law

Queueing Theory

Pollaczek–Khinchine formula

Little's Law

Erlang C Formula



(pic source)

Before 1988 - CISC & page faults

The slowest operation is paging (aka swap)

Performance = # of page faults × # of instructions

RAM is really slow

CPI is unstable

Data Locality is of utmost importance

CISC - Complex Instruction Set Computer - “let’s put all the hard part inside the chip and just use it”

(pic source)

1988-2002 - RISC & Moore's law

CPUs and Instruction set architecture (ISA) became simpler

But compiler grew to be more complex

RAM became larger and cheaper overall

RISC - Let's use fewer instructions, but it will be simple to figure out what happens inside

We've hit several 'walls' and started experimenting with ILP

(pic source)

What limits did we hit?

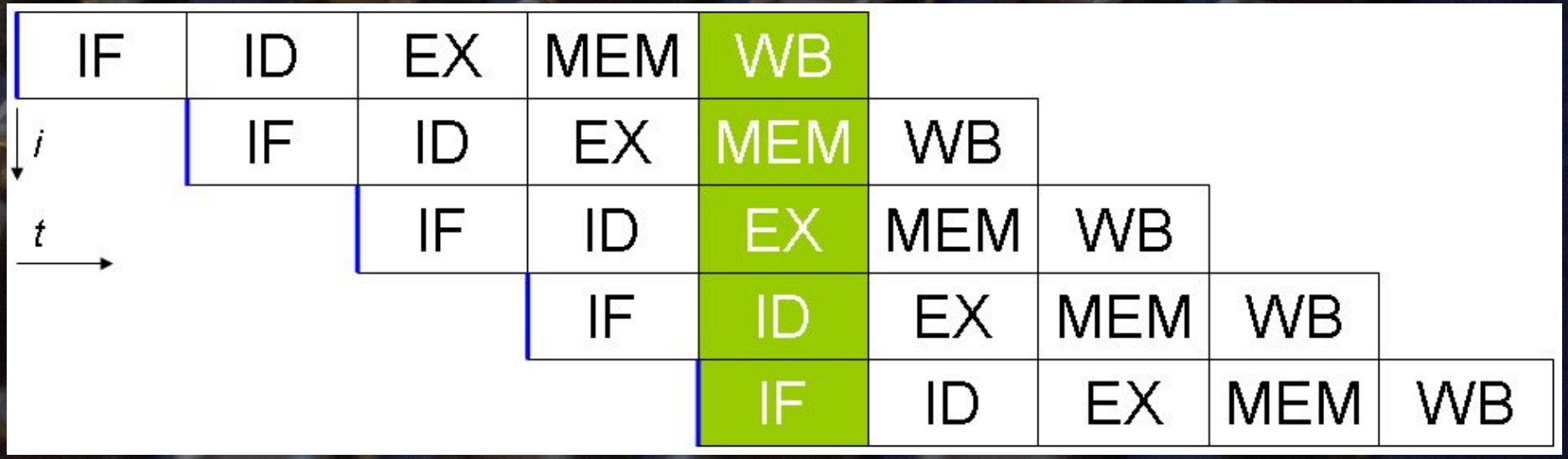
Power supply limit: more frequency -> more power -> more heat -> chip melts into magma

RAM limit - it’s really slow compared to CPU registers

Speed of light: sending a signal through the whole big complex CPU is not an instant operation

(pic source)

Multiple Issue & Pipelining



(pic source)

Branch Prediction & Speculative Execution

Let's guess the result of a condition! (with about 70-80% probability!)

And continue computing instructions until proven guilty (or not!)

What did you say? Meltdown, Spectre? What's that?

(pic source)

ILP Saga

Register Renaming

Out-of-Order Execution

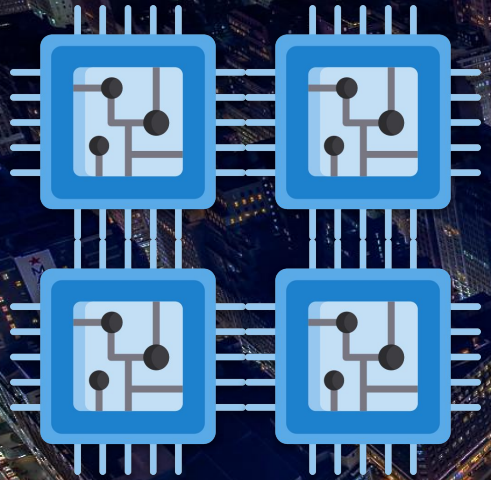
Instruction Prefetching

No-Lockup

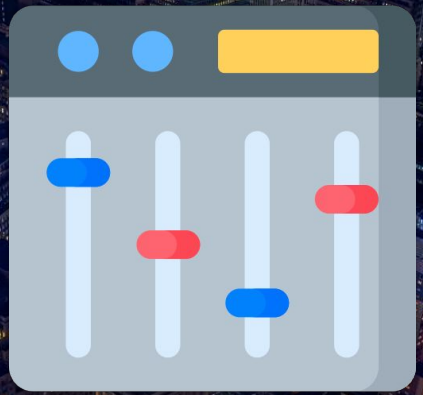
Cliff Click - A Crash Course In Modern Hardware

(pic source)

2002 - ...



More cores



More abstraction layers

Performance is dominated by cache misses

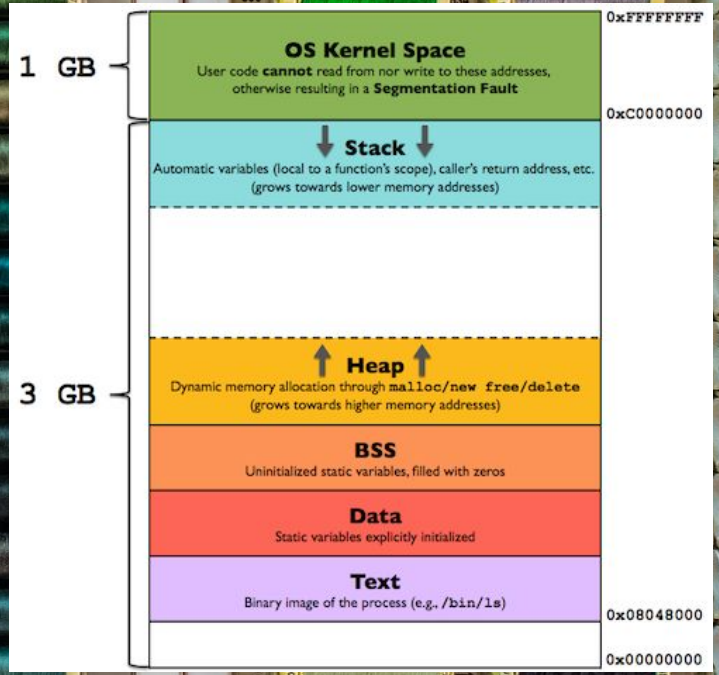
Icons made by Freepik from www.flaticon.com

(pic source)

Mind the Layout

Emery Berger - Performance Matters & V2!

Producing Wrong Data Without Doing Anything Obviously Wrong! (paper)



(pic source)

(pic source)

Hardware

- Both program and data it works with is located in RAM, just in different parts of it
- CPU runs instructions in an order, one by one sequentially
- Any program can ask CPU to read any memory address and do something with it

```
00000000        push    ebp
00000001        mov     ebp, esp
00000003        movzx   ecx, [ebp+arg_0]
00000007        pop     ebp
00000008        movzx   dx, cl
0000000C        lea     eax, [edx+edx]
0000000F        add     eax, edx
00000011        shl     eax, 2
00000014        add     eax, edx
00000016        shr     eax, 8
00000019        sub     cl, al
0000001B        shr     cl, 1
0000001D        add     al, cl
0000001F        shr     al, 5
00000022        movzx   eax, al
00000025        retn
```

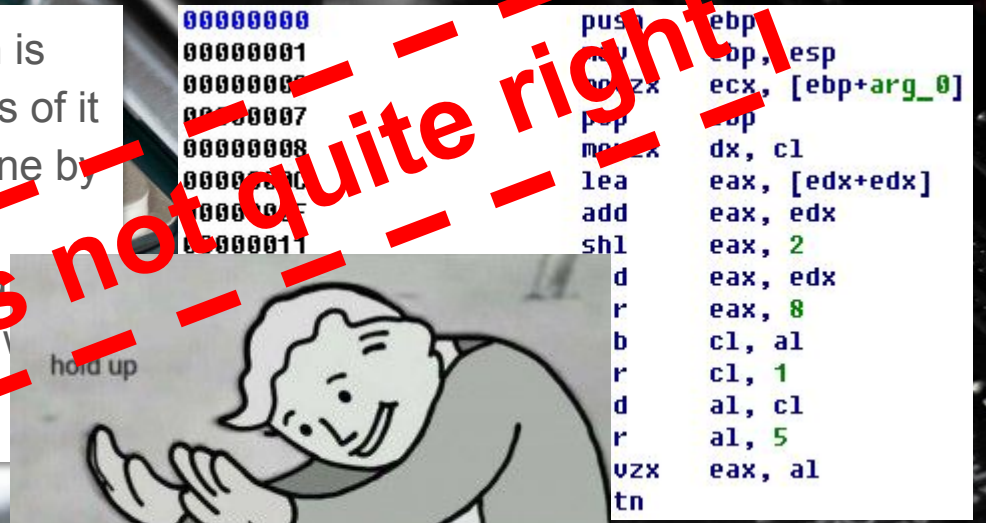
(pic source)

(pic source)

Hardware

- Both program and data it works with is located in RAM, just in different parts of it
- CPU runs instructions in an order, one by one sequentially
- Any program can ask CPU to read a memory address and do something with it

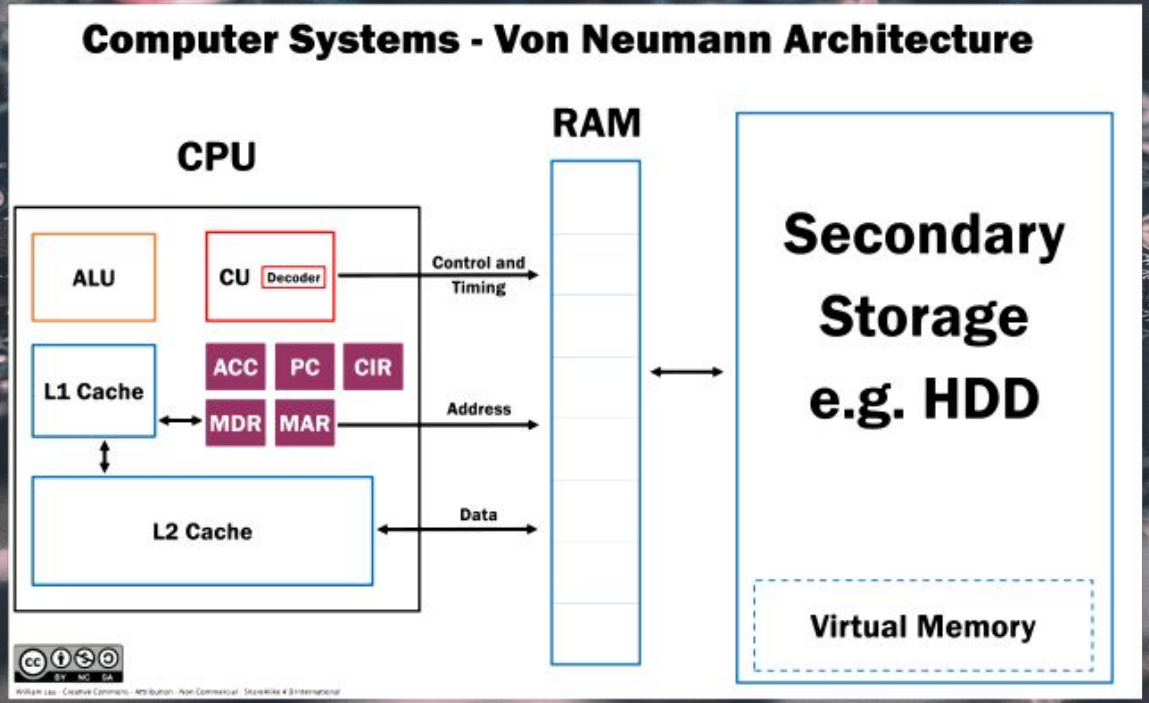
"Wait, that's not quite right"



(pic source)

(pic source)

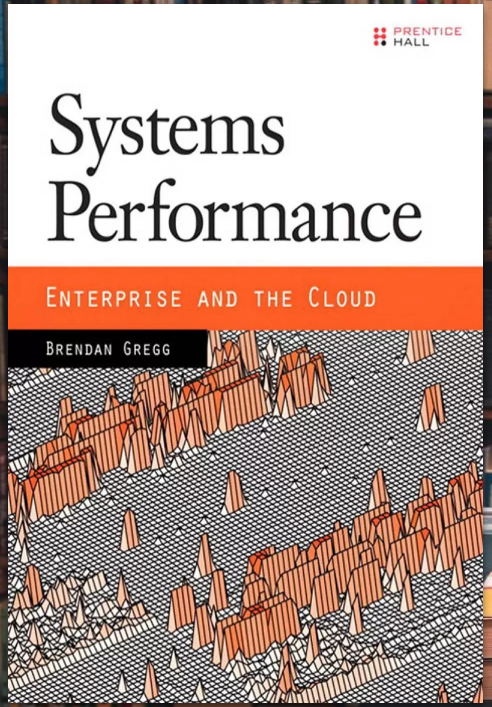
von Neumann Architecture

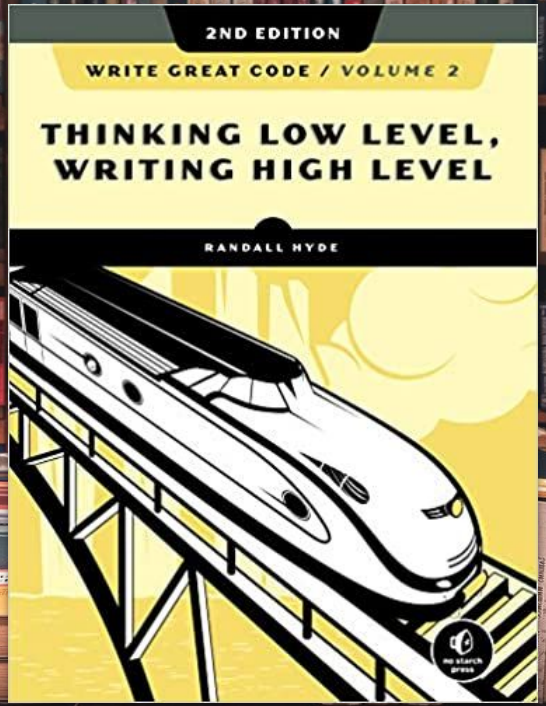


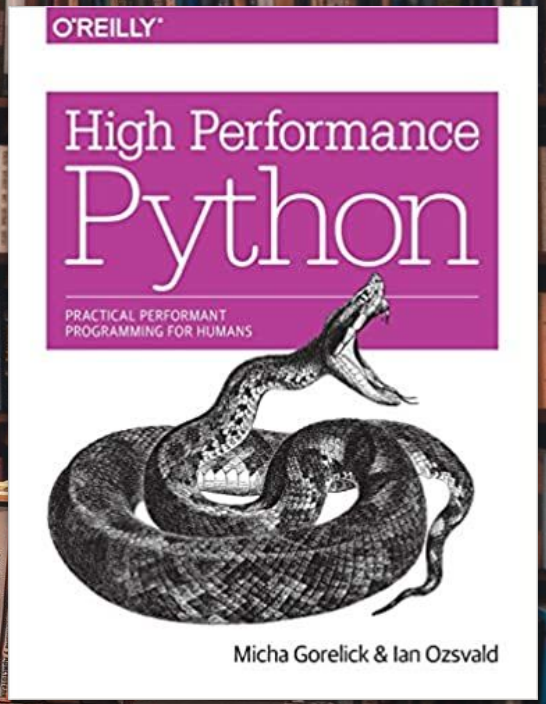
(pic source)

(pic source)

Read & Watch







(pic source)

Read & Watch - 2

Eli Bendersky - Adventures in JIT Compilation (4 parts)

Bpftrace

Linux Monitoring (rus)

How to deal with Pandas (rus)

Kavya Joshi - Practical look at performance theory

Conway's Law

Julia Evans - Profiling and tracing with Perf

Floating-point arithmetic

(pic source)

Questions?

https://t.me/enchantner

https://twitter.com/enchantner

https://www.linkedin.com/in/nickmarkov/

(pic source)