

# High Performance Python

Joe



# Introduction



# Afternoon Objectives

1. Compare and Contrast processes & threads, and discuss possible times to implement one or another.
2. Implement both in code to start to understand failure points for either.

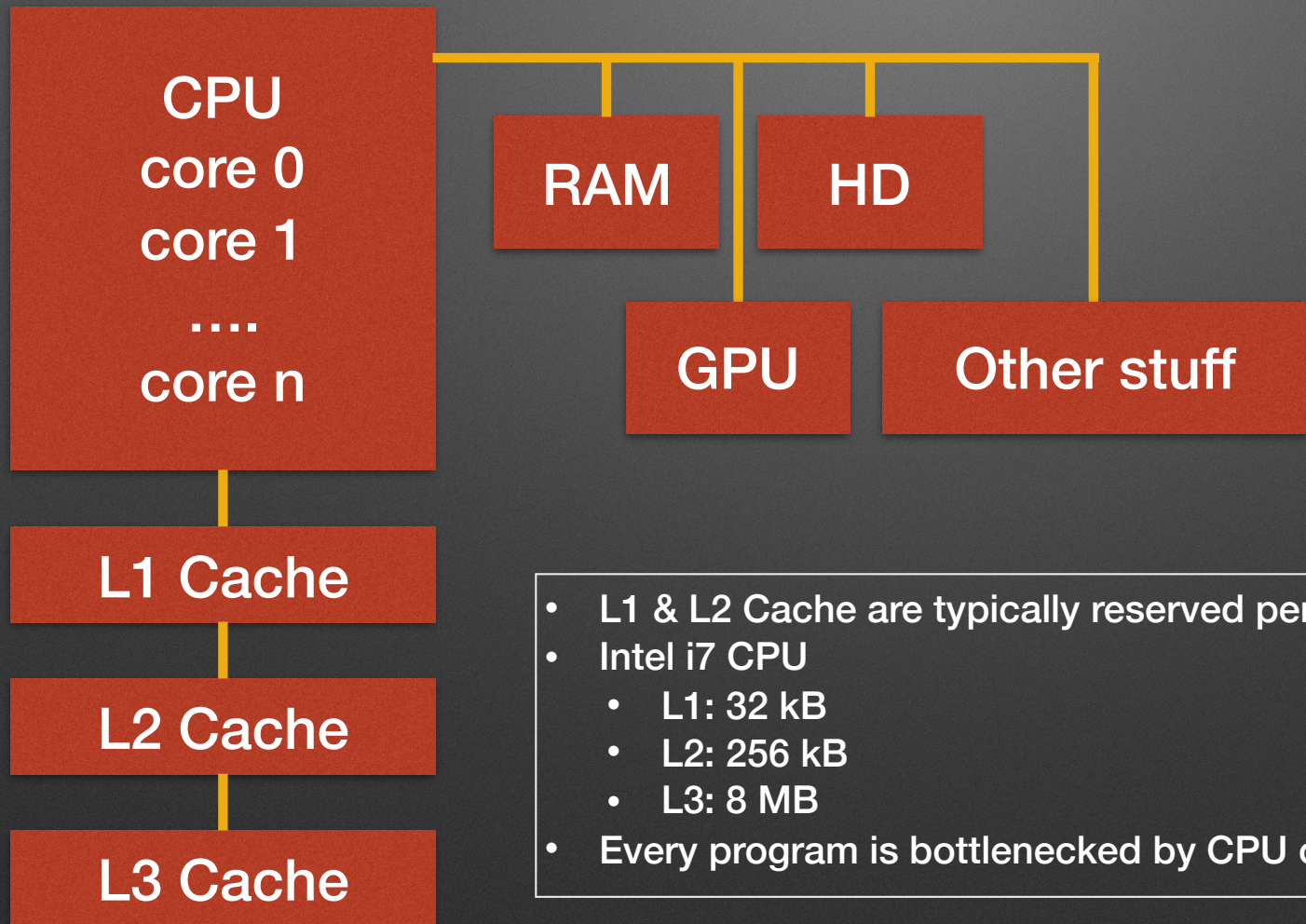
Credit for the programs goes to Ryan Henning



# Computation Limitations



# Your Computer



- L1 & L2 Cache are typically reserved per core, while L3 is shared
- Intel i7 CPU
  - L1: 32 kB
  - L2: 256 kB
  - L3: 8 MB
- Every program is bottlenecked by CPU or IO



# Python

## Threads vs Processes

### Process

An instance of a computer program that is being executed

Each process has it's own memory, program text, ect, and as such can run on any core.

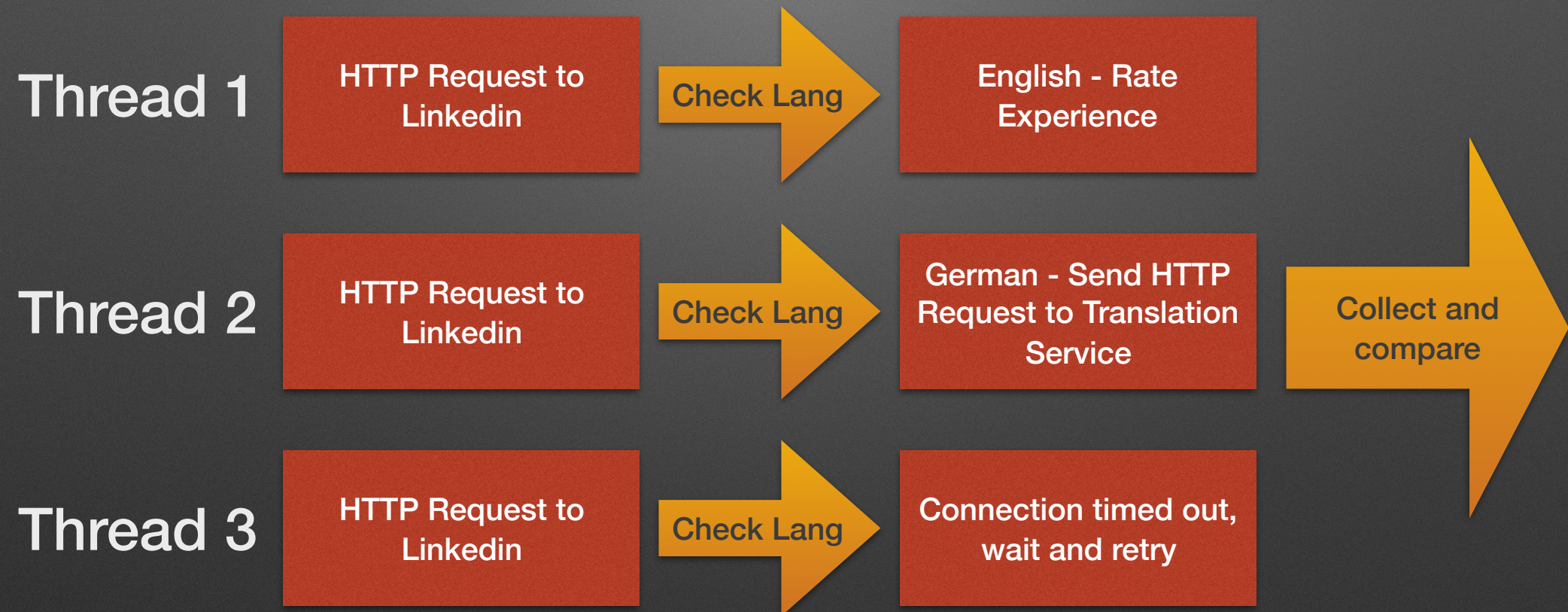
### Thread

A Process is composed of one or more threads of execution.

Threads can run concurrently, however, python is not thread safe (i.e. memory can be overwritten), and has a Global Interpreter Lock that prevents multiple threads from running at once.



# Thread Example





# Afternoon Objectives

1. Compare and Contrast processes & threads, and discuss possible times to implement one or another.
2. Implement both in code to start to understand failure points for either.

Credit for the programs goes to Ryan Henning