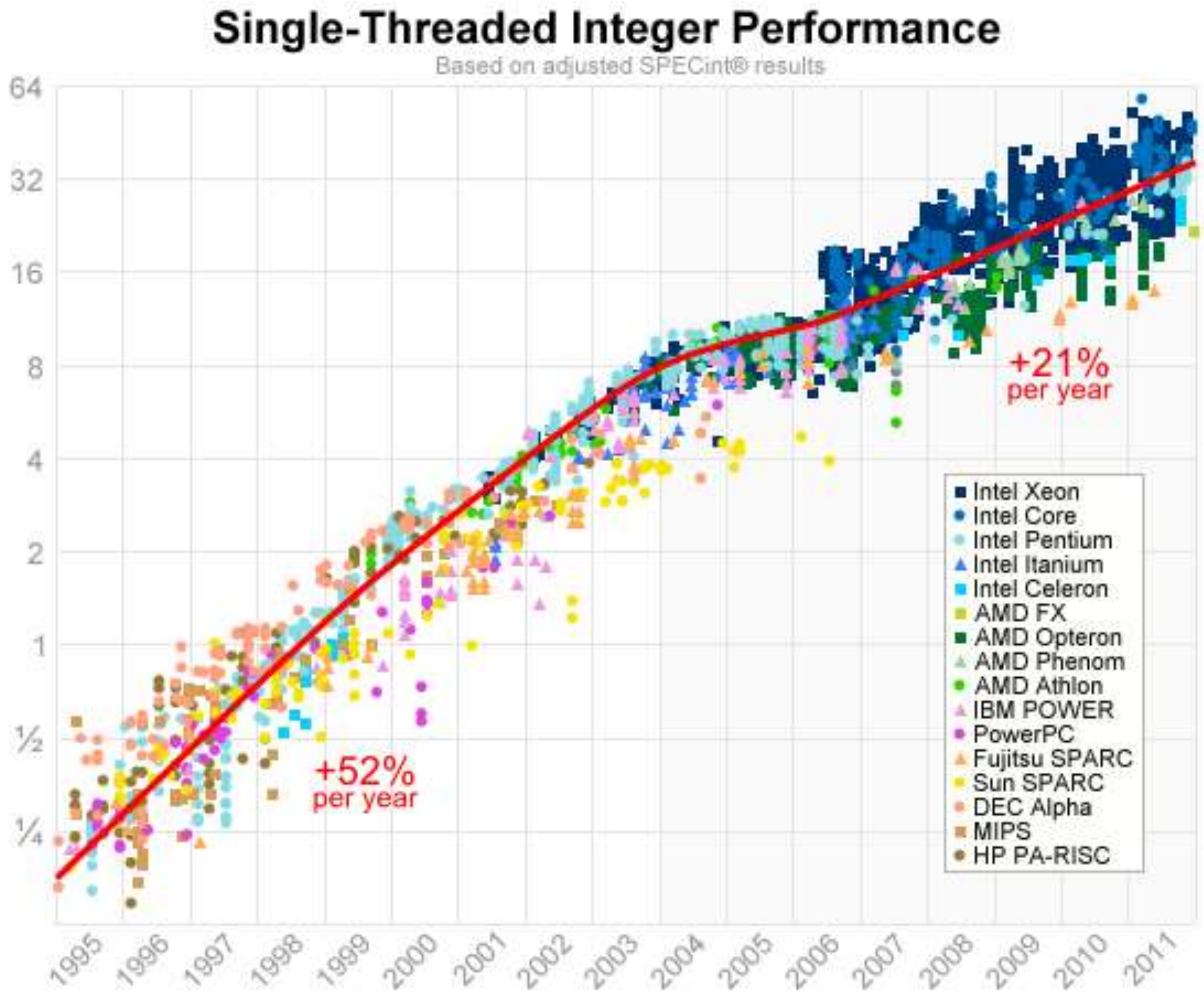
Await and tasks from the ground up

Jiří Činčura

www.tabsoverspaces.com

@cincura_net



Source: preshing.com

Current state of CPUs

- Free lunch is over
  - 22nm => 50 silicon atoms
  - Cooling, frequency
- Smarter processors
  - Hyperthreading
- More cores

Coarse-grained parallelism

- Processes are expensive
- Threads are expensive
- A lot of threads ➔ context switching

- No cake for developers?

ThreadPool

- OK-ish solution for 2018
  - Limited features
  - A lot of manual work required

- Tasks to saves us?

Tasks

- Task Parallel Library
- Rich API
- Easy to use (and even easier with async/await)
- Uses ThreadPool
  - Usually ☺

CPU bound vs I/O bound code

- Threads and tasks vs asynchronous I/O
    - Overlapped I/O
- Asynchronous
    - No blocking
        - Scalability
    - Concurrency
- For I/O there’s no “background thread”

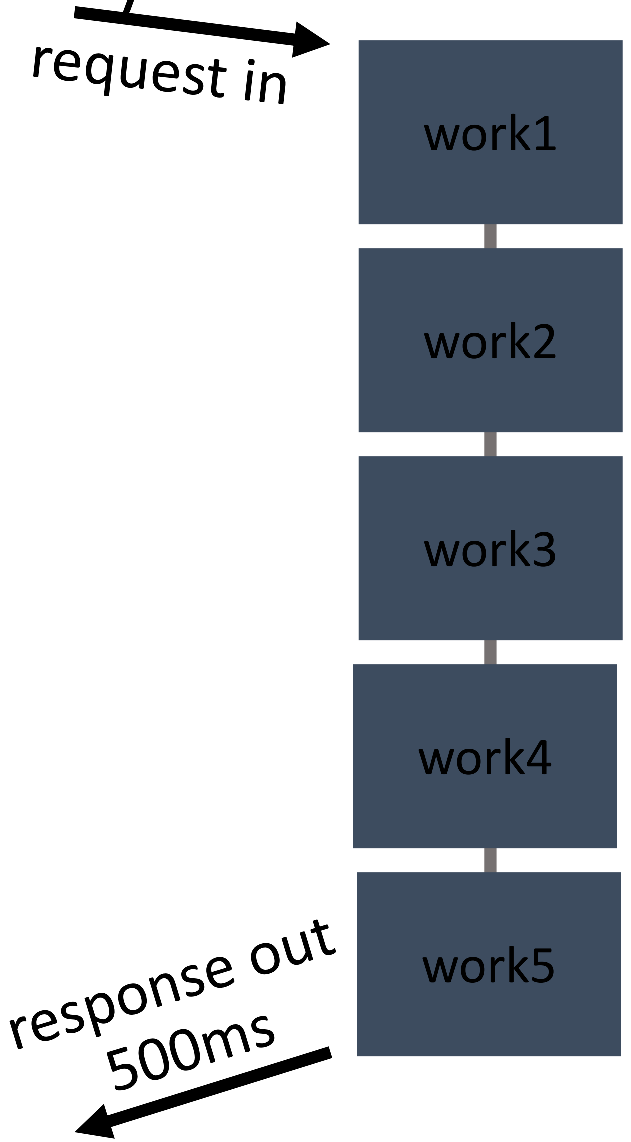
There’s no “background thread”?

- My code ➔ BCL ➔ OS/Kernel ➔ IRP
- ISR ➔ DPC ➔ APC ➔ IOCP

CPU bound vs I/O bound code

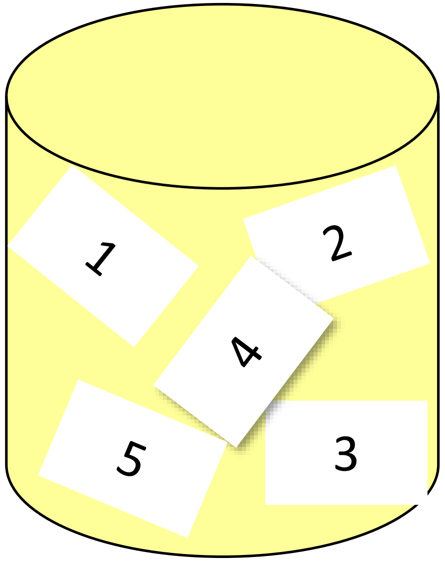
```
public List<Something> LoadSomething()
{
  var result = new List<Something>();
  for (var i = 1; i <= 5; i++)
  {
    var s = Something.LoadFromNetwork(id: i);
    result.Add(s);
  }
  return result;
}
```

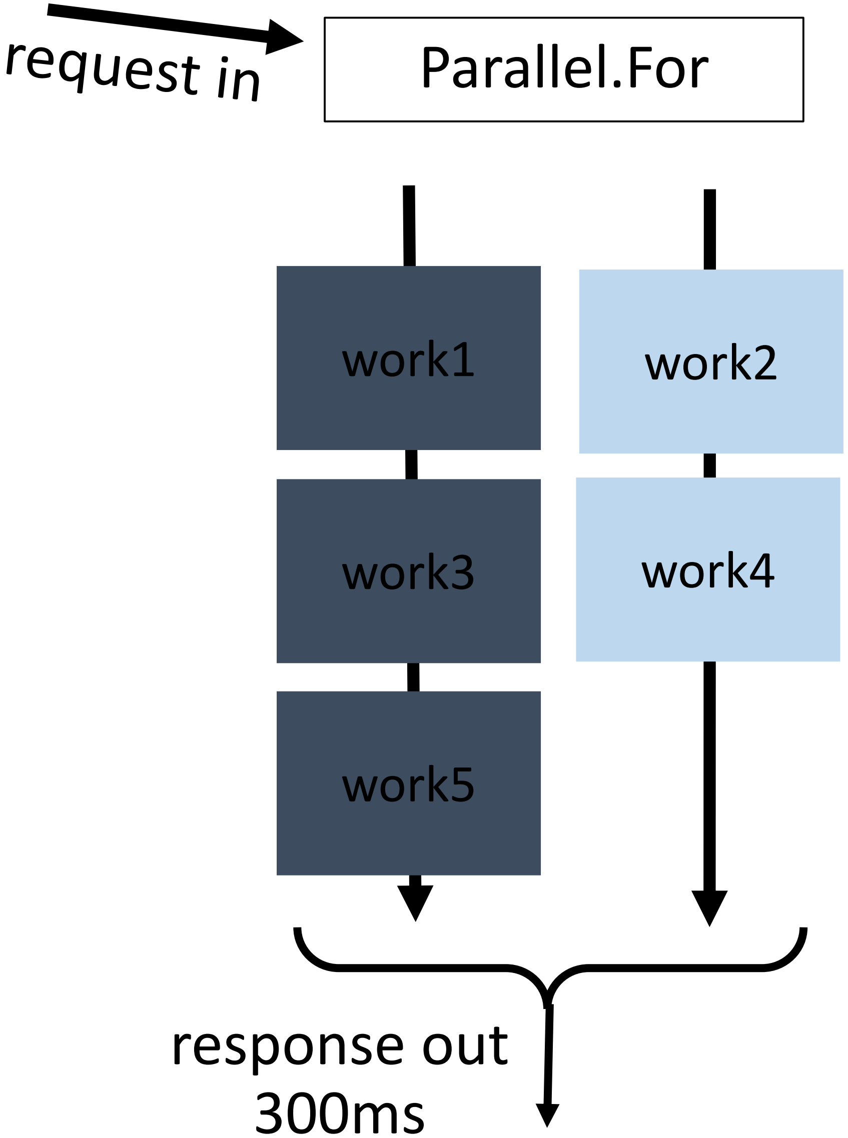
CPU bound vs I/O bound code



Source: Lucian Wischik

CPU bound vs I/O bound code




request in

Parallel.For

work1

work2

work3

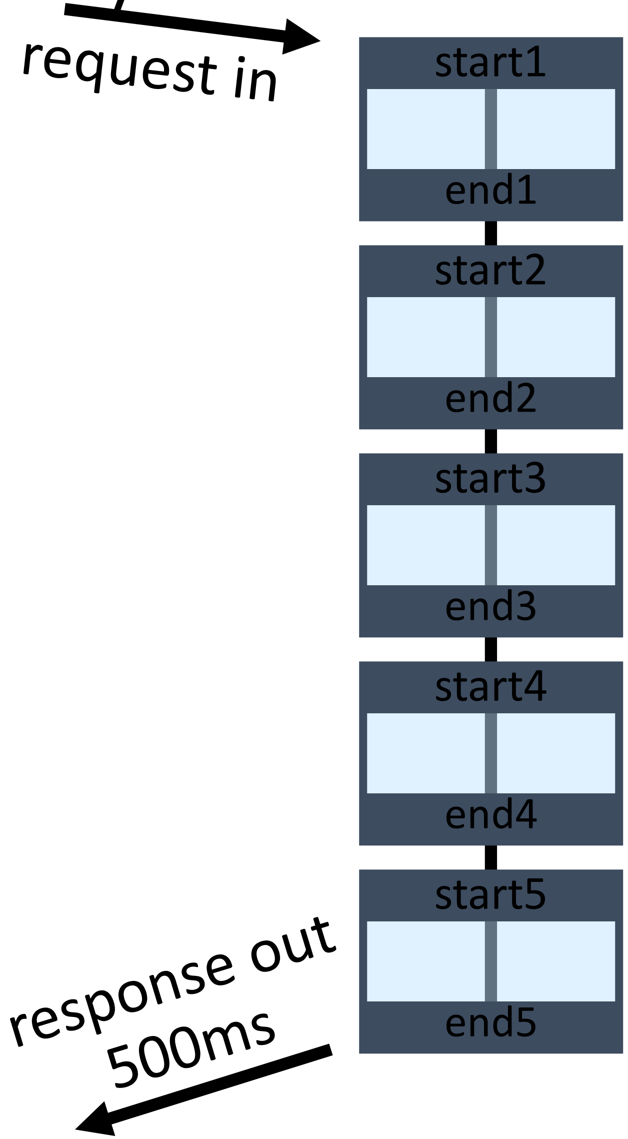
work4

work5

response out
300ms

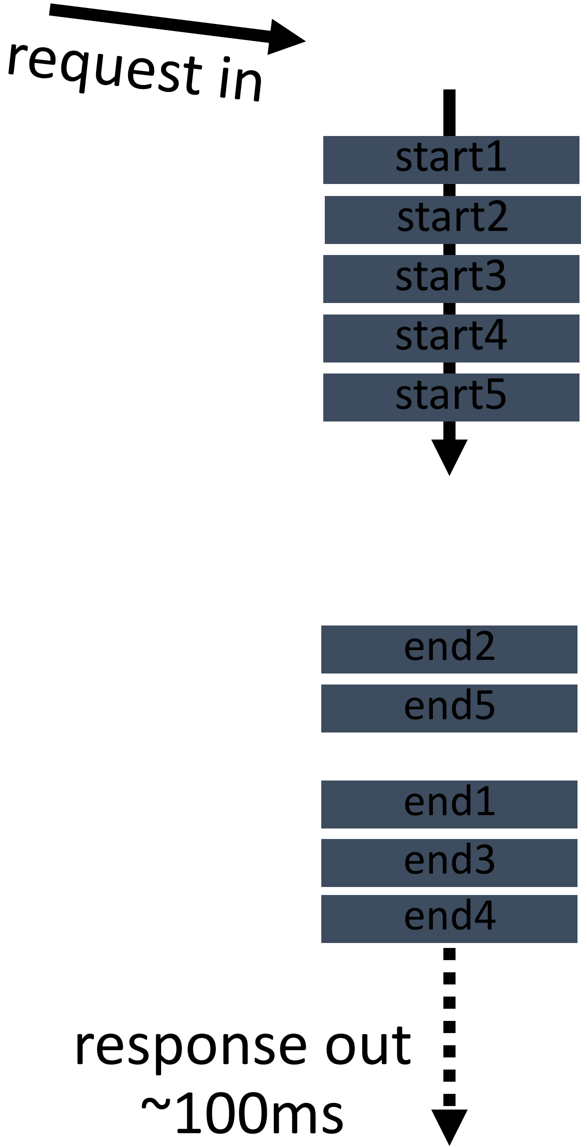
Source: Lucian Wischik

CPU bound vs I/O bound code



Source: Lucian Wischik

CPU bound vs I/O bound code


request in

start1

start2

start3

start4

start5

end2

end5

end1

end3

end4

response out
~100ms

Source: Lucian Wischik

Before async/await

- Asynchronous Programming Model
    - BeginXxx, EndXxx

- Try to read stream…

Async/await

- Simpler code for callbacks
  - Compiler solves the plumbing
  - State machine (similar to IEnumerable<T>)
  - For-loops, usings, try-catch blocks, …
- But could be hard to master
  - i.e. deadlocks, performance degradation
- Try to read stream v2…

Async/await

- Works basically on Task/Task<T>/ValueTask<T>
- CPU or I/O bound
- CPU bound tasks
  - Delegate tasks
- I/O bound tasks
  - Promise tasks

Q & A