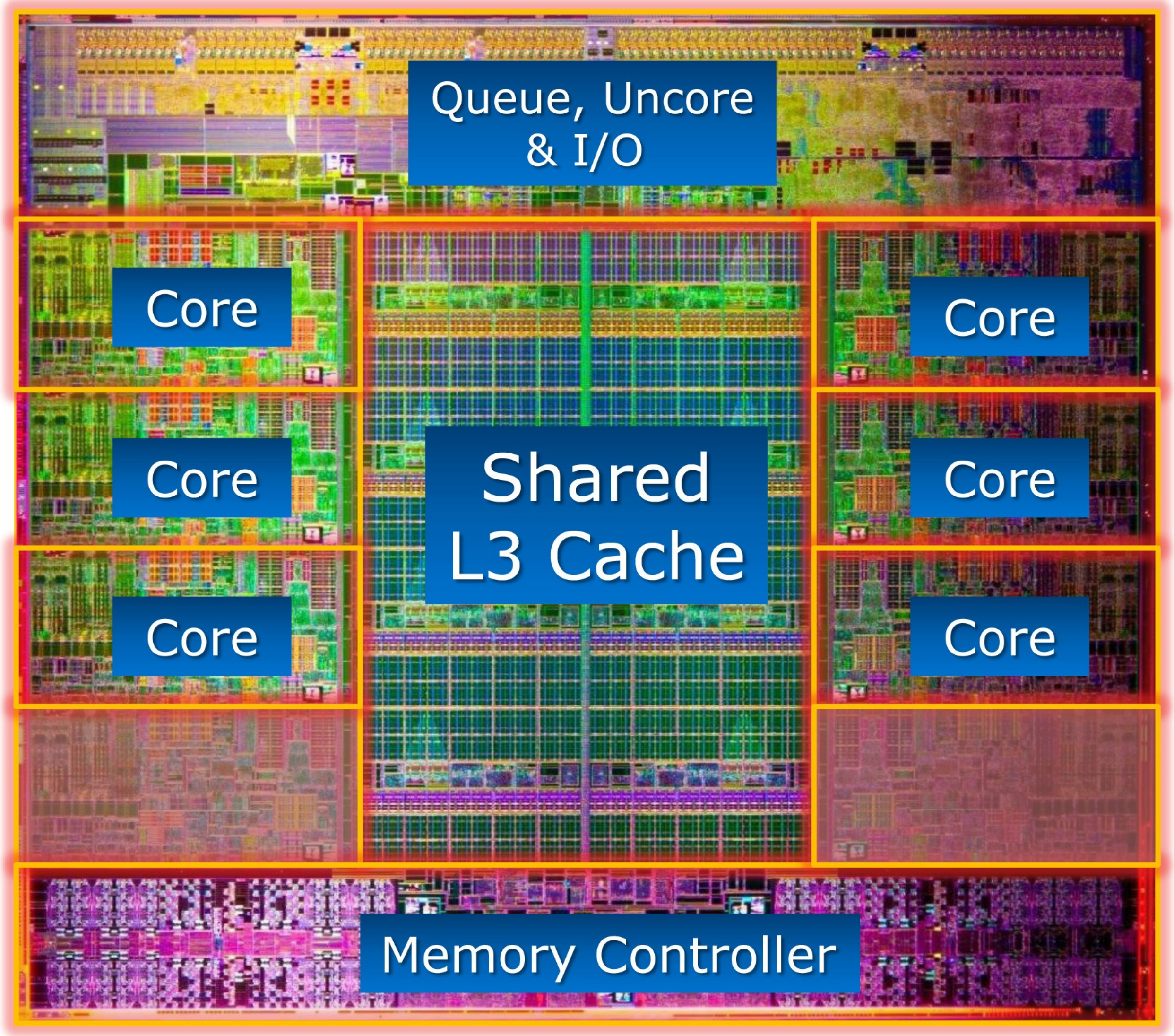
deal.II Users and Developers Training

March 21 – 24 2016

Timo Heister (heister@clemson.edu)
Luca Heltai (luca.heltai@sissa.it)

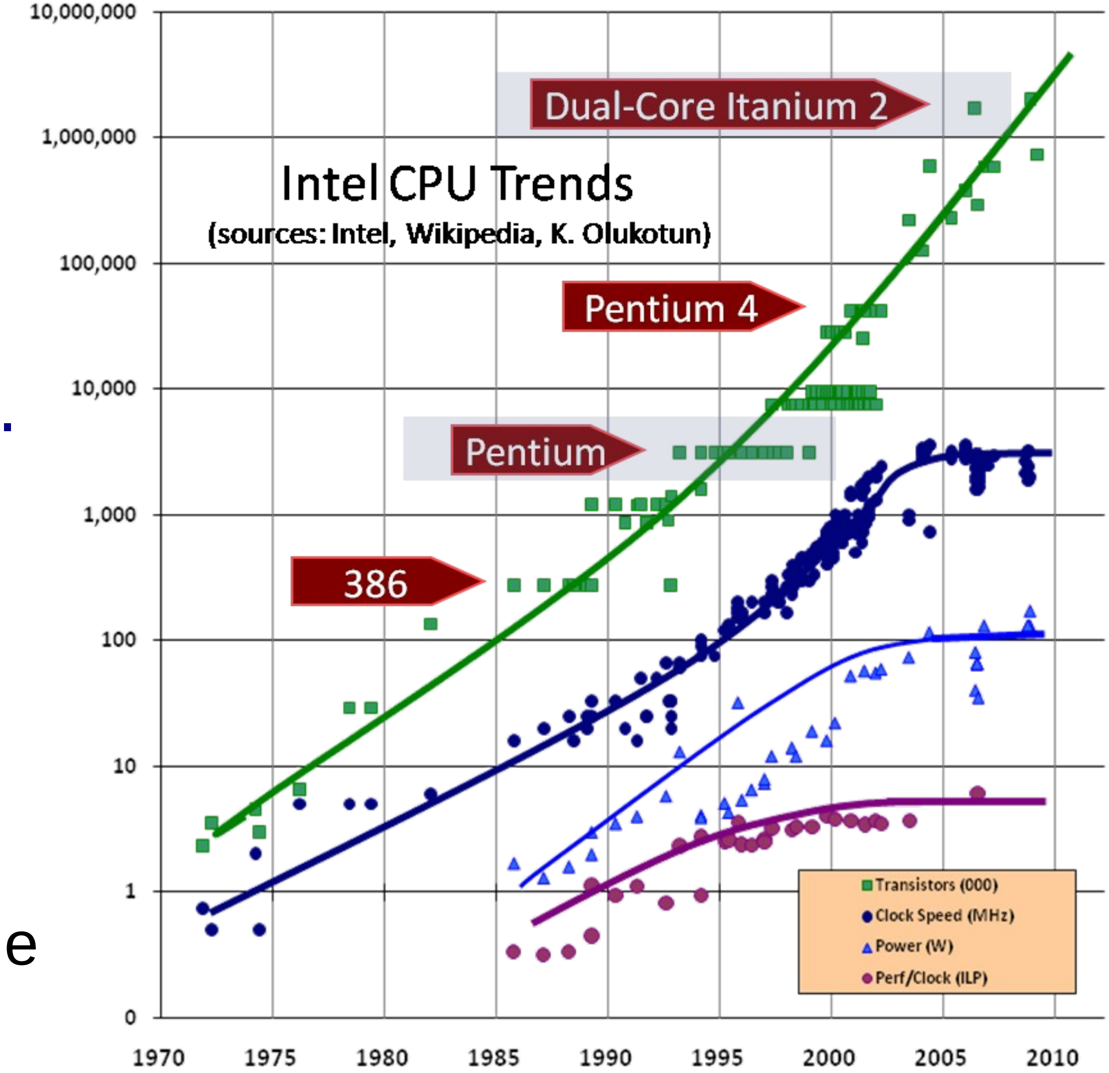
Parallel Computing: Introduction



A modern CPU: Intel Core i7

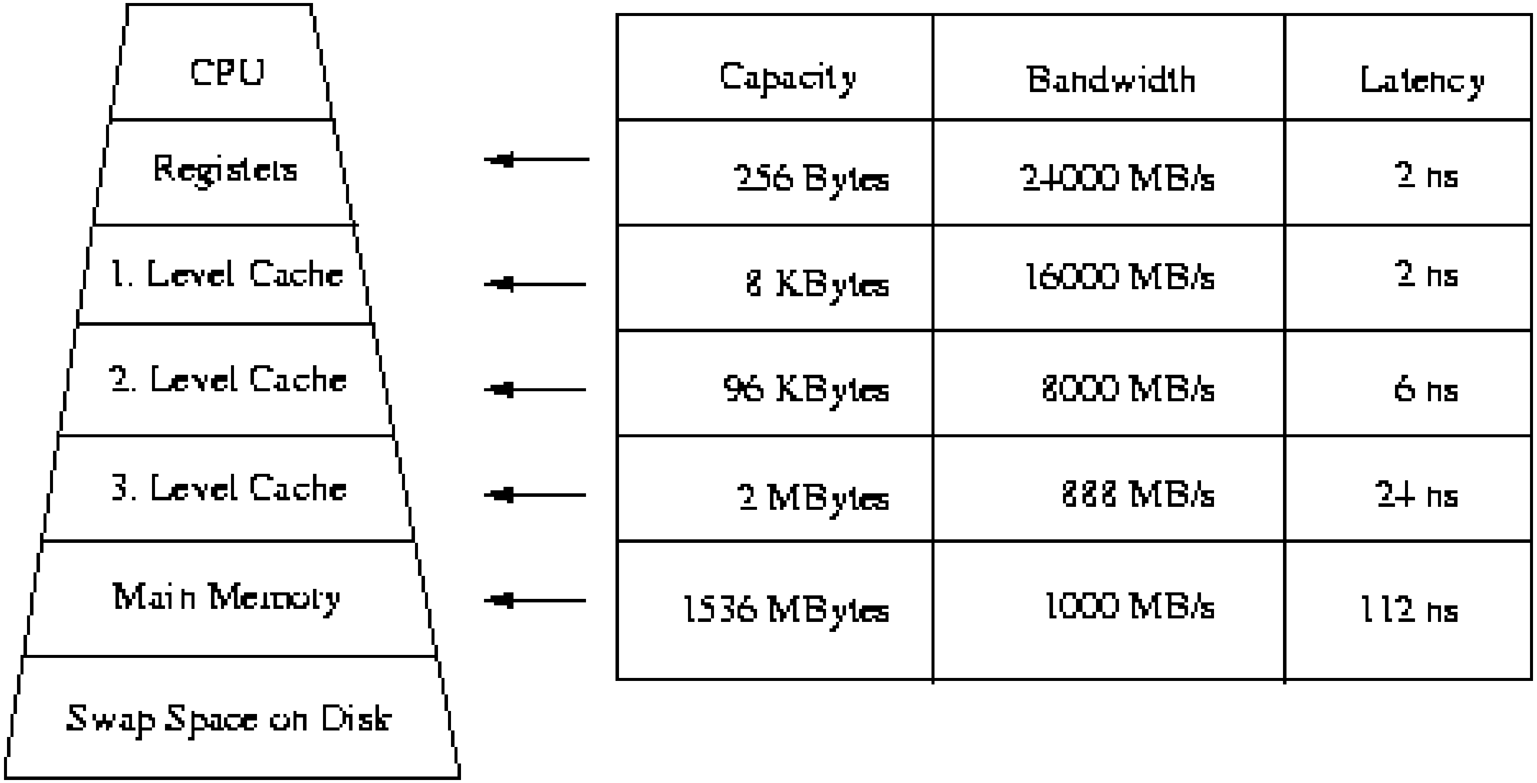
Basics

- Single cores are not getting (much) faster
- “the free lunch is over”: http://www.gotw.ca/publications/concurrency-ddj.htm
- Concurrency is only option:
  - SIMD/vector instructions
  - Several cores
  - Several chips in one node
  - Combine nodes into supercomputer



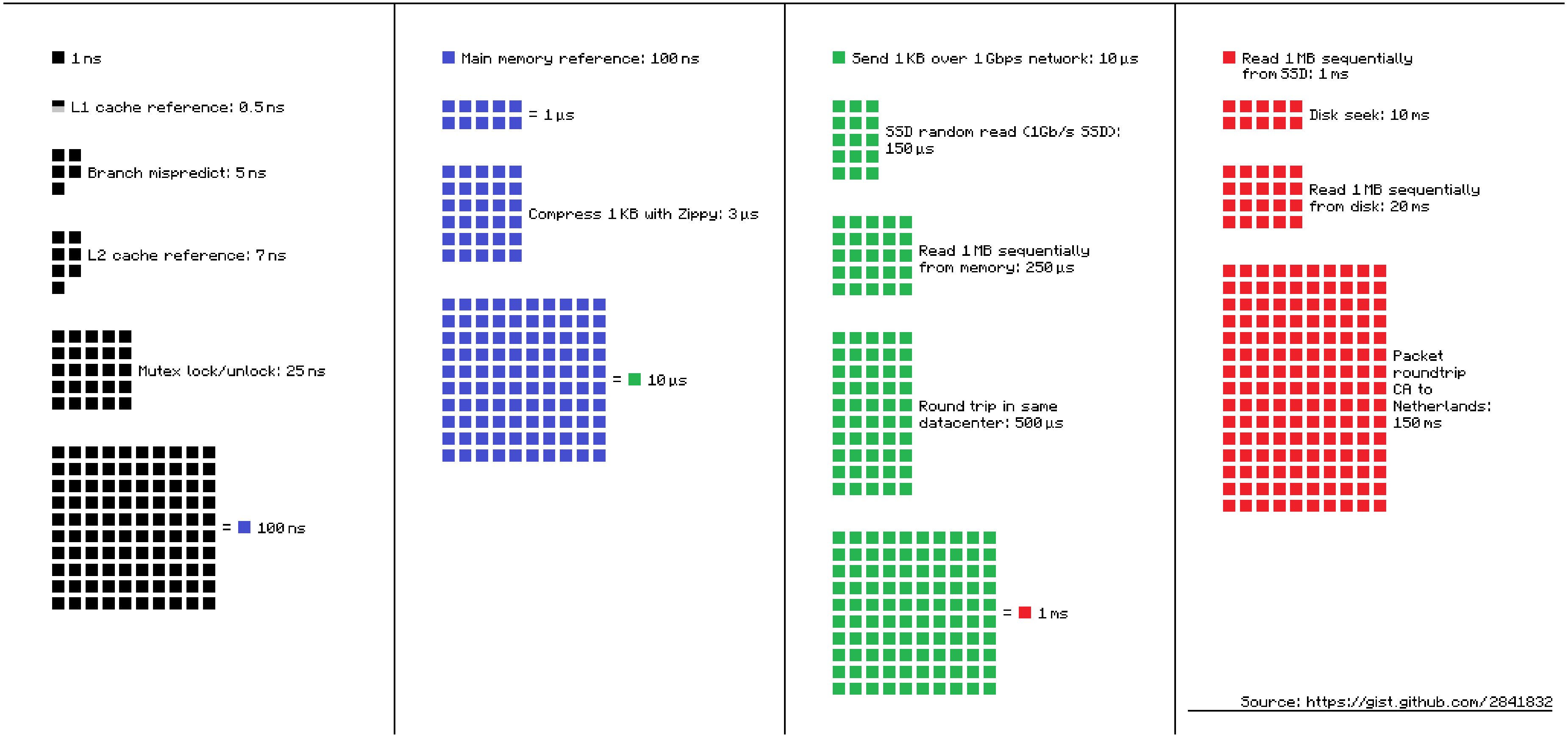
Hierarchy of memory

- Latency: time CPU gets data after requesting
- Bandwidth: how much data per second?
- prefetching of data, “cache misses” are expensive
- automatically managed by processor



Capacity	Bandwidth	Latency
256 Bytes	24000 MB/s	2 ns
8 KBytes	16000 MB/s	2 ns
96 KBytes	8000 MB/s	6 ns
2 MBytes	888 MB/s	24 ns
1536 MBytes	1000 MB/s	112 ns

Latency Numbers Every Programmer Should Know



Source: https://gist.github.com/2841832

https://gist.github.com/hellerbarde/2843375

Amdahl's Law

- Task: serial fraction s, parallel fraction $p=1-s$
- N workers (whatever that means)
- Runtime: $T(N) = (1-s)T(1)/N + sT(1)$
- Speedup $T(1)/T(N)$, N to infinity:

  max_speedup = 1/ s
- http://en.wikipedia.org/wiki/Amdahl%27s_law
- Reality: $T(N) = (1-s)T(1)/N + sT(1) + aN + bN^2$

Summary

- Computing much faster than memory access
- Parallel computing required: no free lunch!
- Communication is serial fraction (or worse when increasing with N!)
- Communication in Amdahl's law is main challenge in parallel computing