# LOW LATENCY C++

## C++ ON SEA 2019

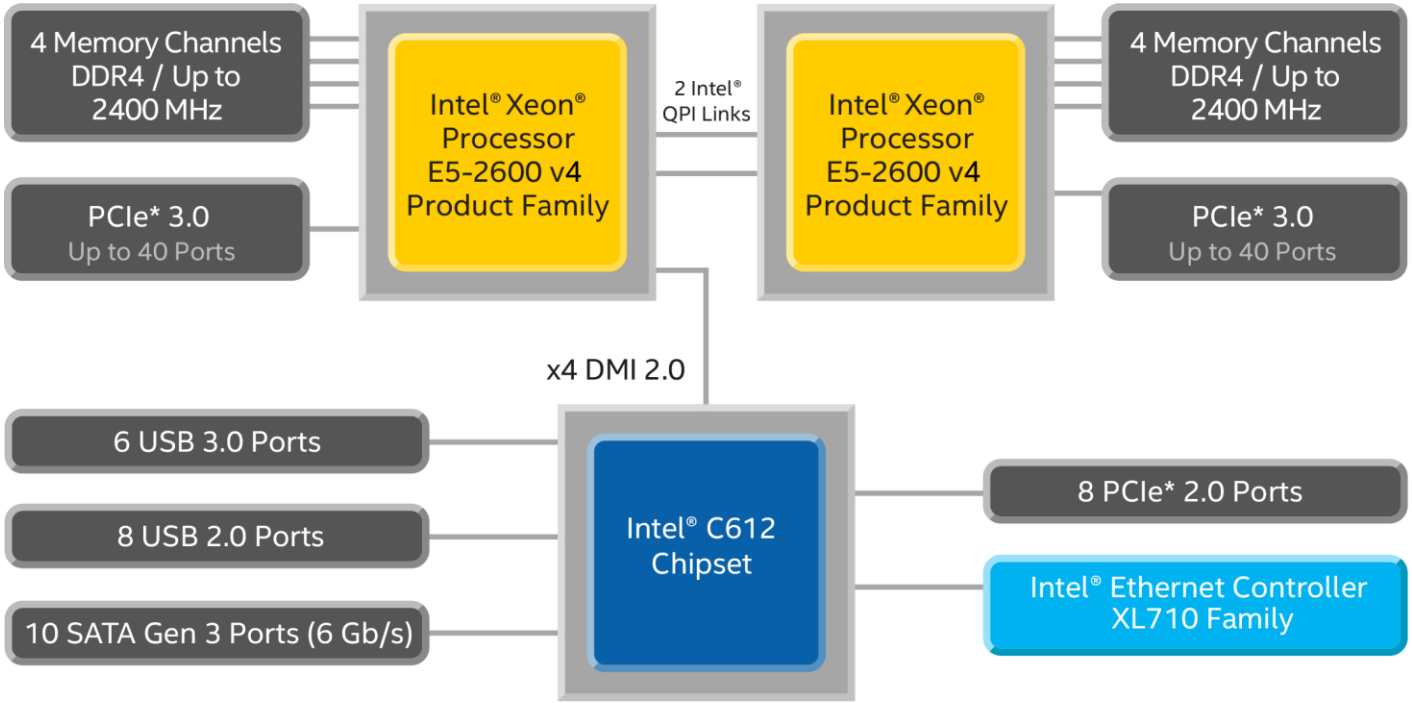
DAVID GROSS
WALTHER ZWART

06/02/2019

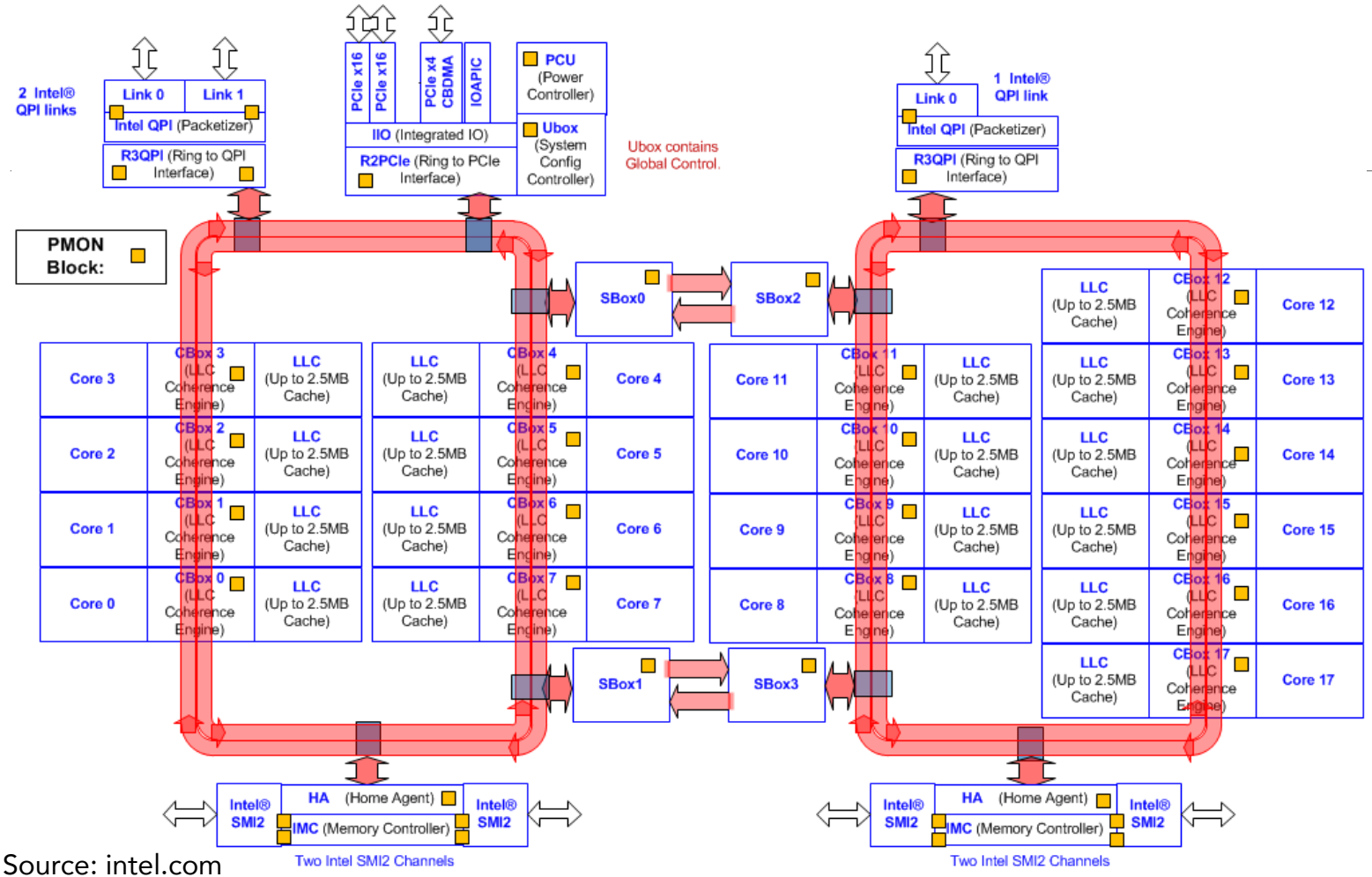
# CPU AND MEMORY ARCHITECTURE

C++ ON SEA 2019

# MODERN X86 ARCHITECTURE




Source: intel.com




Source: intel.com

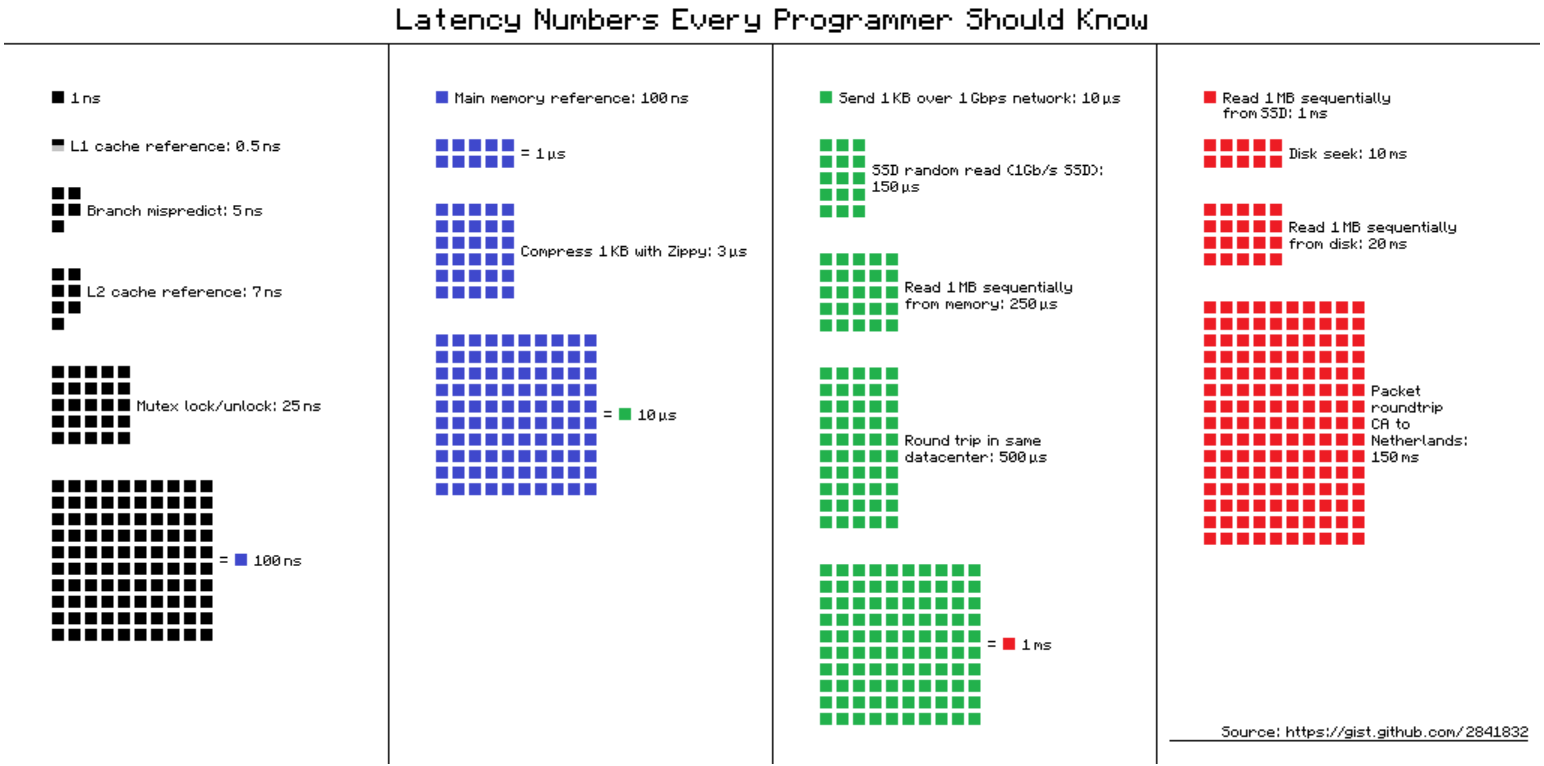
# MODERN X86 ARCHITECTURE

- Growing importance of the Uncore
  - Integrated memory controller (iMC)
  - Processor point-to-point Interconnect, e.g. Intel QuickPath Interconnect (QPI)
  - Snooping, e.g. Caching and Home agents
  - Last Level Cache (LLC)
  - On-die GPU
  - IOs controllers, e.g. PCIe

# CPU CACHES LAYOUT

# KNOW YOUR LATENCIES

# KNOW YOUR LATENCIES



# BENCHMARKING C++

C++ ON SEA 2019

# WHY BENCHMARKING?

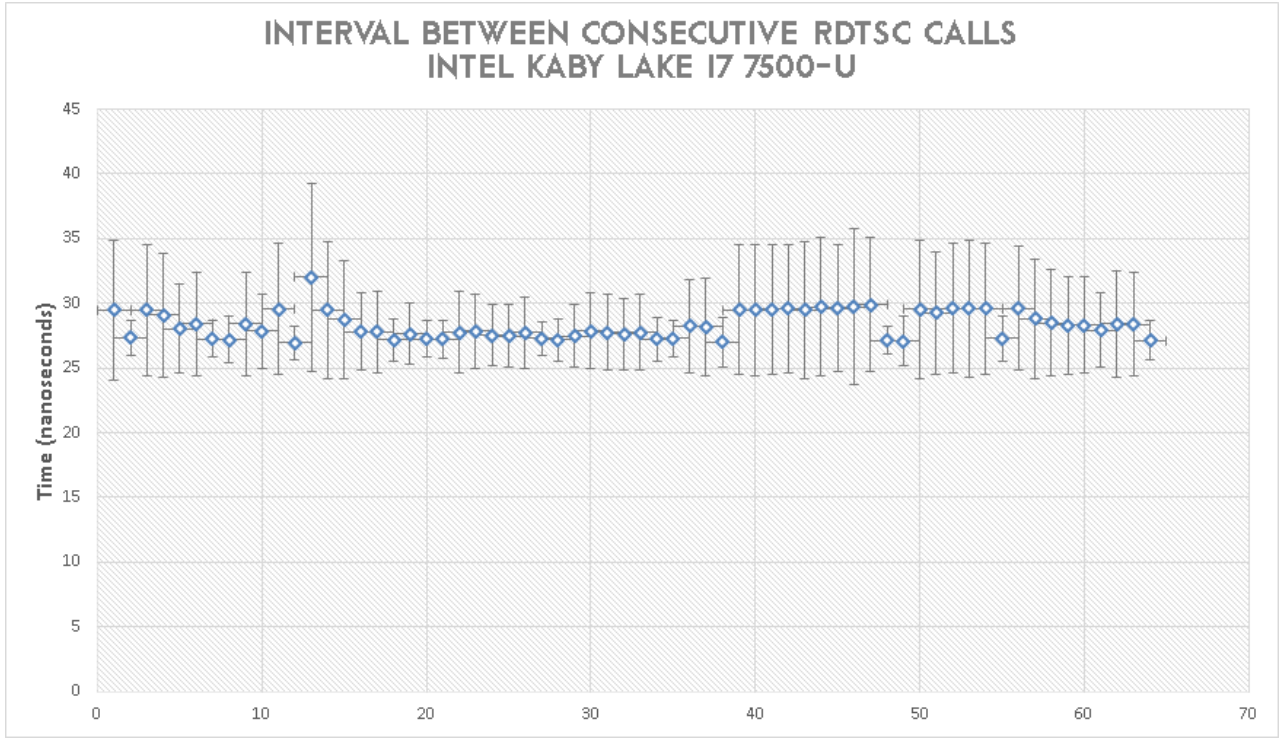
# WHY BENCHMARKING?

- If you don't benchmark your code, you don't care about its performance
- Iterative process
  - Measure: macro benchmarking, metrics, …
  - Find : Pareto principle
  - Isolate and optimize: micro benchmarking
  - Repeat

# LET'S BENCHMARK

C++ ON SEA 2019

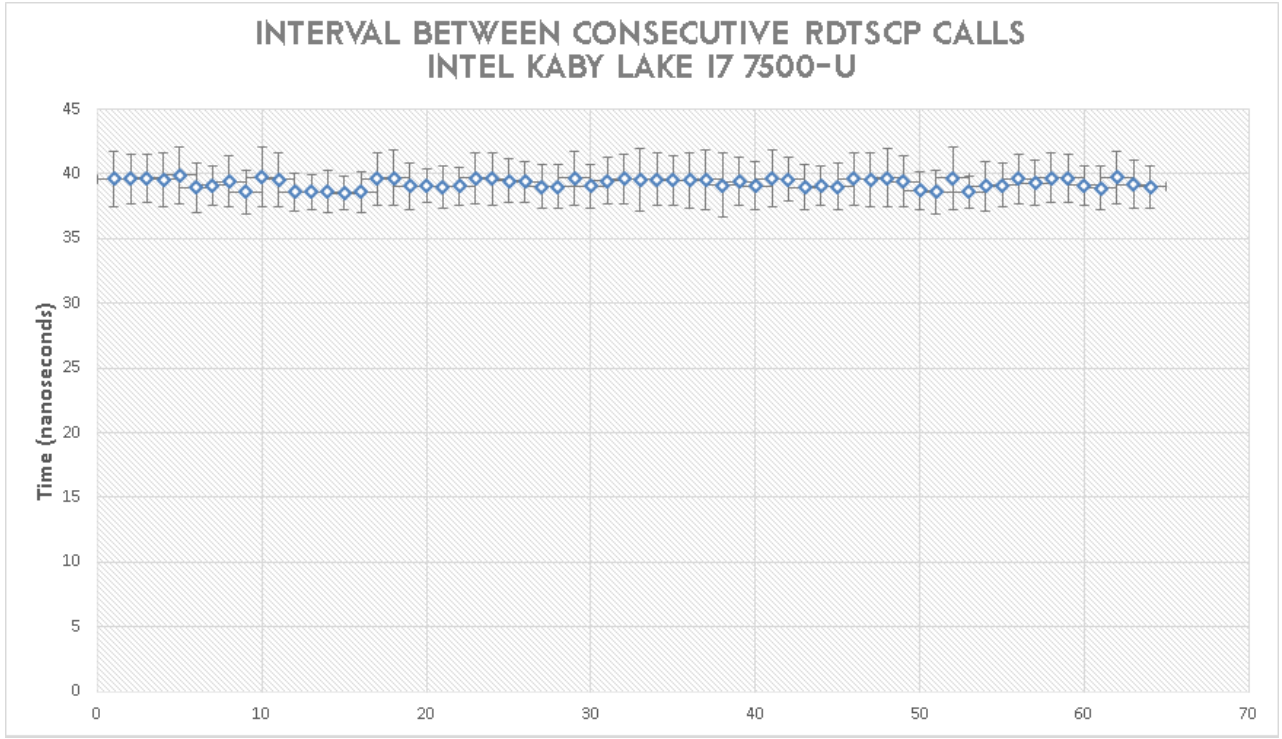
# RDTSC, RDTSCP? FENCE, NO FENCE?




1 sample: 32-256 calls

Sample std dev: 3.49ns

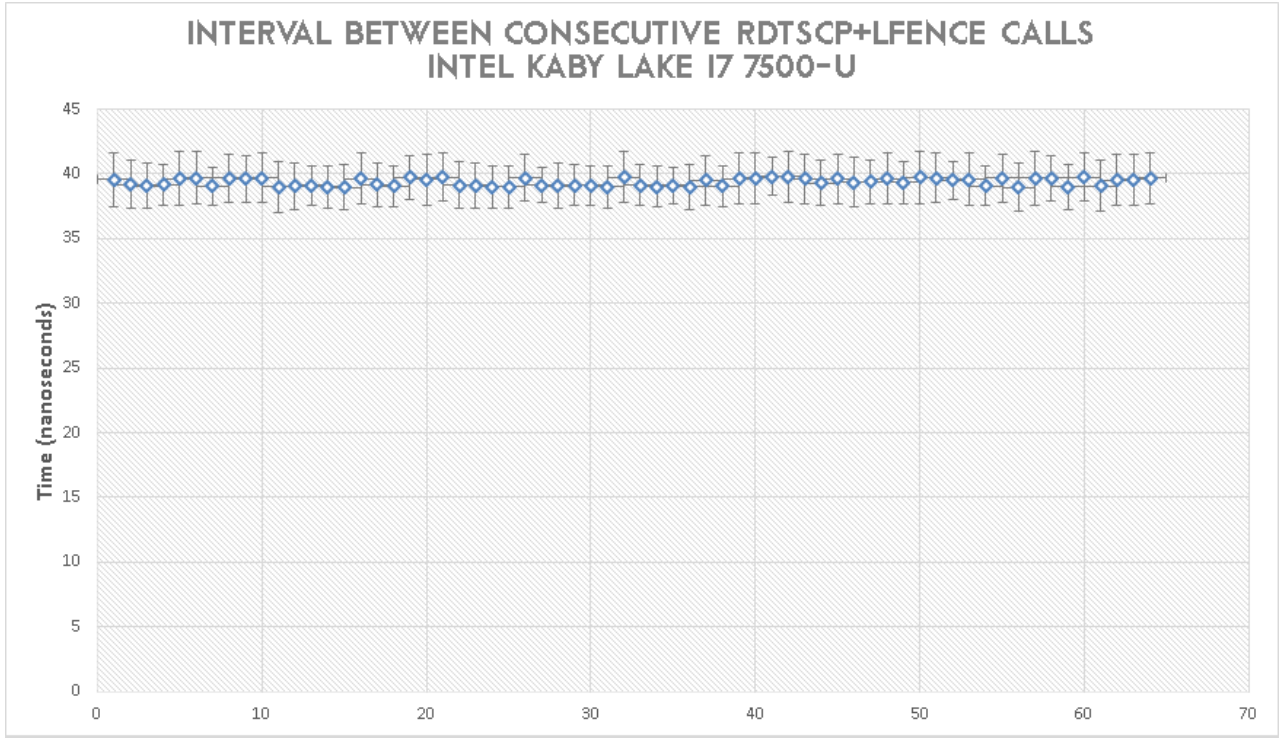
# RDTSC, RDTSCP? FENCE, NO FENCE?




1 sample: 32-256 calls

Sample std dev: 1.85ns

# RDTSC, RDTSCP? FENCE, NO FENCE?




1 sample: 32-256 calls

Sample std dev: 1.72ns