GPU Teaching Kit

Accelerated Computing

Lecture 1.3 – Course Introduction

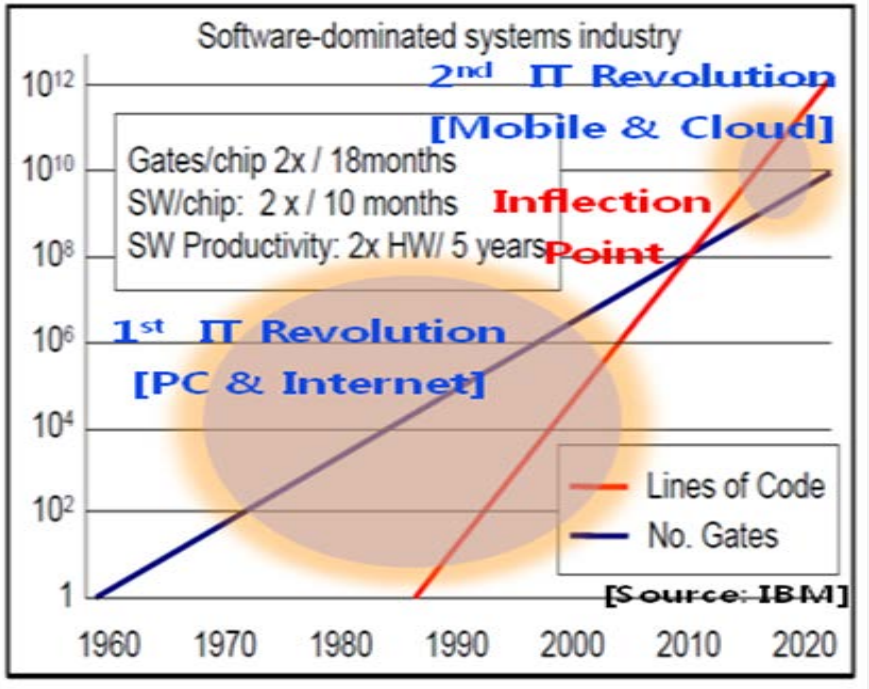
Portability and Scalability in Heterogeneous Parallel Computing

Objectives

– To understand the importance and nature of scalability and portability in parallel programming

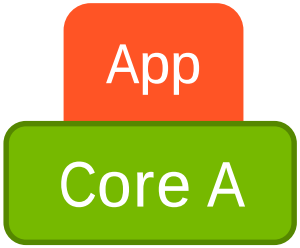
Software Dominates System Cost

- SW lines per chip increases at 2x/10 months
- HW gates per chip increases at 2x/18 months
- Future systems must minimize software redevelopment



Keys to Software Cost Control

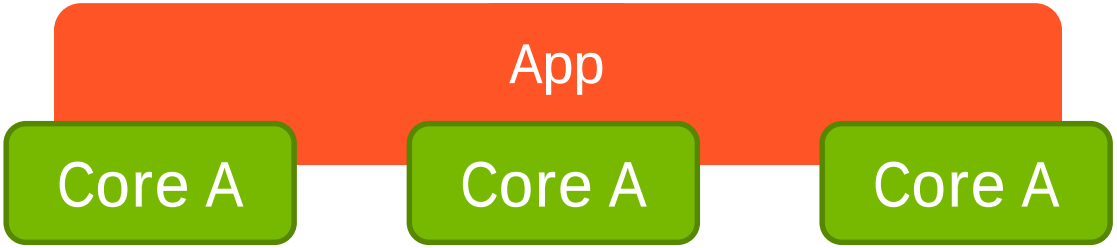
- Scalability



Keys to Software Cost Control

- Scalability
  - **The same application runs efficiently on new generations of cores**

Keys to Software Cost Control

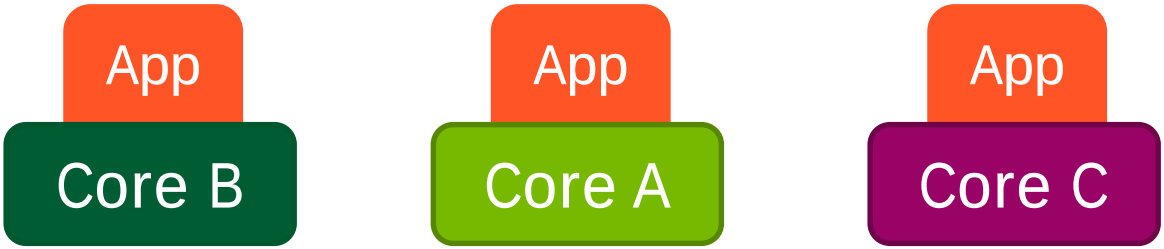


- Scalability
  - The same application runs efficiently on new generations of cores
  - **The same application runs efficiently on more of the same cores**

More on Scalability

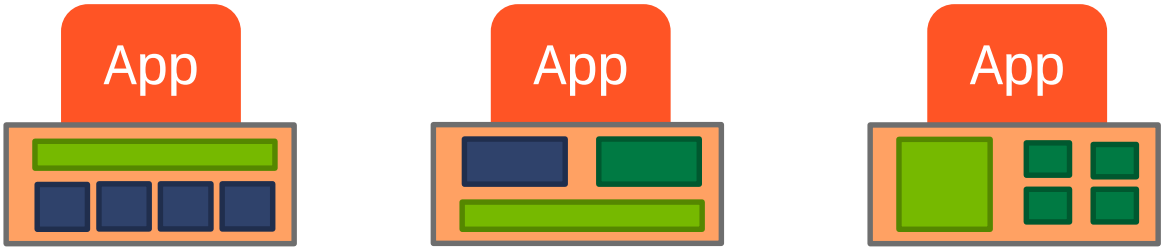
- Performance growth with HW generations
  - Increasing number of compute units (cores)
  - Increasing number of threads
  - Increasing vector length
  - Increasing pipeline depth
  - Increasing DRAM burst size
  - Increasing number of DRAM channels
  - Increasing data movement latency

Keys to Software Cost Control



- Scalability
- **Portability**
  - The same application runs efficiently on different types of cores

Keys to Software Cost Control



- Scalability
- Portability
  - The same application runs efficiently on different types of cores
  - The same application runs efficiently on systems with different organizations and interfaces

More on Portability

- Portability across many different HW types
  - Across ISAs (Instruction Set Architectures) - X86 vs. ARM, etc.
  - Latency oriented CPUs vs. throughput oriented GPUs
  - Across parallelism models - VLIW vs. SIMD vs. threading
  - Across memory models - Shared memory vs. distributed memory

GPU Teaching Kit

Accelerated Computing