BC3.04a Introduction to HPC Programming

Tran Giang Son, tran-giang.son@usth.edu.vn

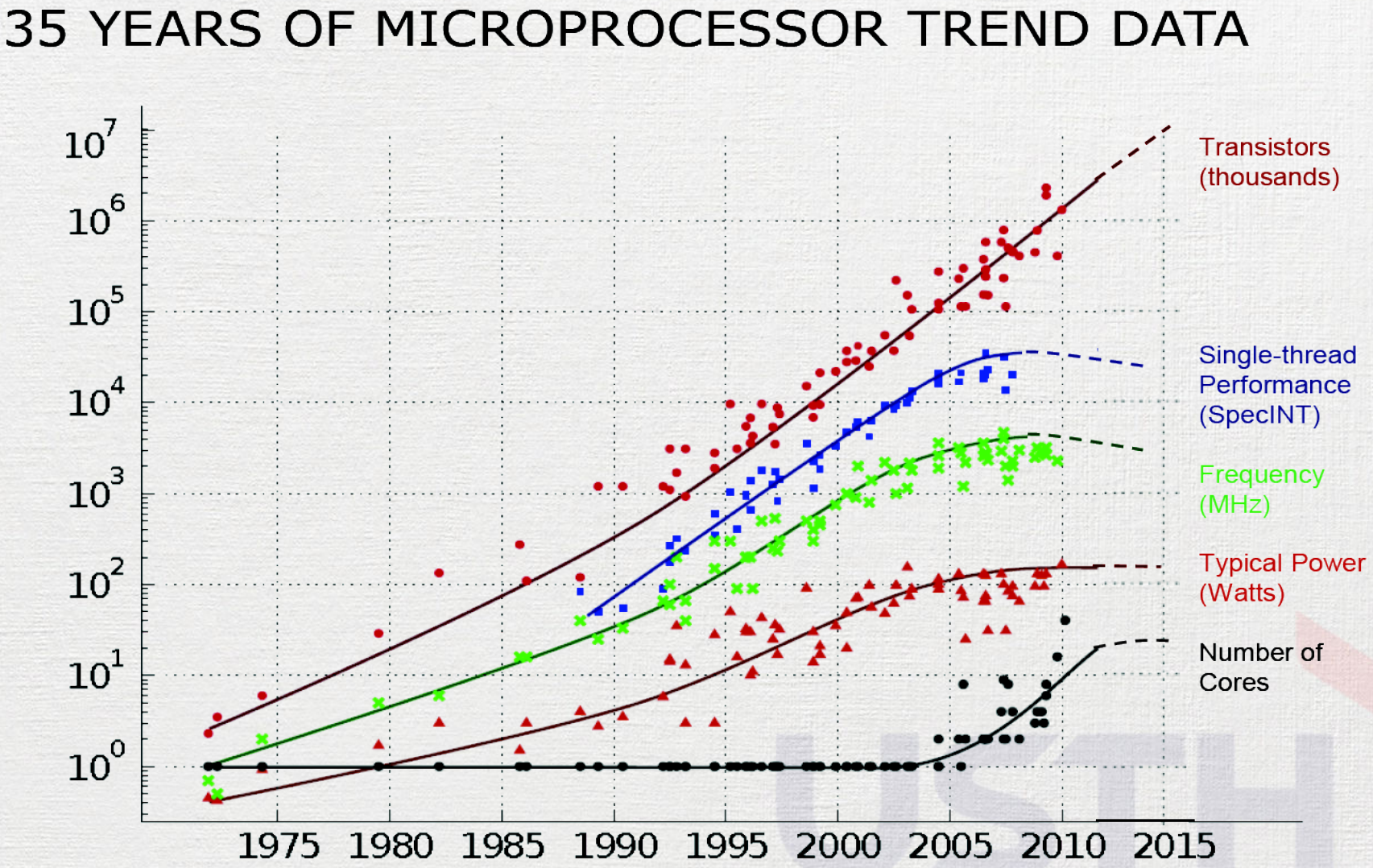
ICT Department, USTH

> *For over a decade prophets have voiced ... single computer has reached its limits and that truly significant advances can be made only by interconnection of a multiplicity of computers...*
>
> - Gene M. Amdahl, IBM, 1967

Distributed Architecture

Centralized Computing

- One big, fast guy



Distributed Systems

- Hardware and software of a collection of independent computers
- Cooperate to implement some functionality

Distributed Systems

- Network
  - Internet
  - Wide area network
  - Intranet
- Example
  - Web apps: Facebook, Twitter...
  - Cloud-based systems
  - Scientific applications: SETI@Home, Folding@Home

Parallel System?

- Multiprocessor systems
  - Direct access to shared memory, UMA
  - Interconnection network
- Multicomputer parallel systems
  - No direct access to shared memory, NUMA
  - Easier scalability

Parallel System

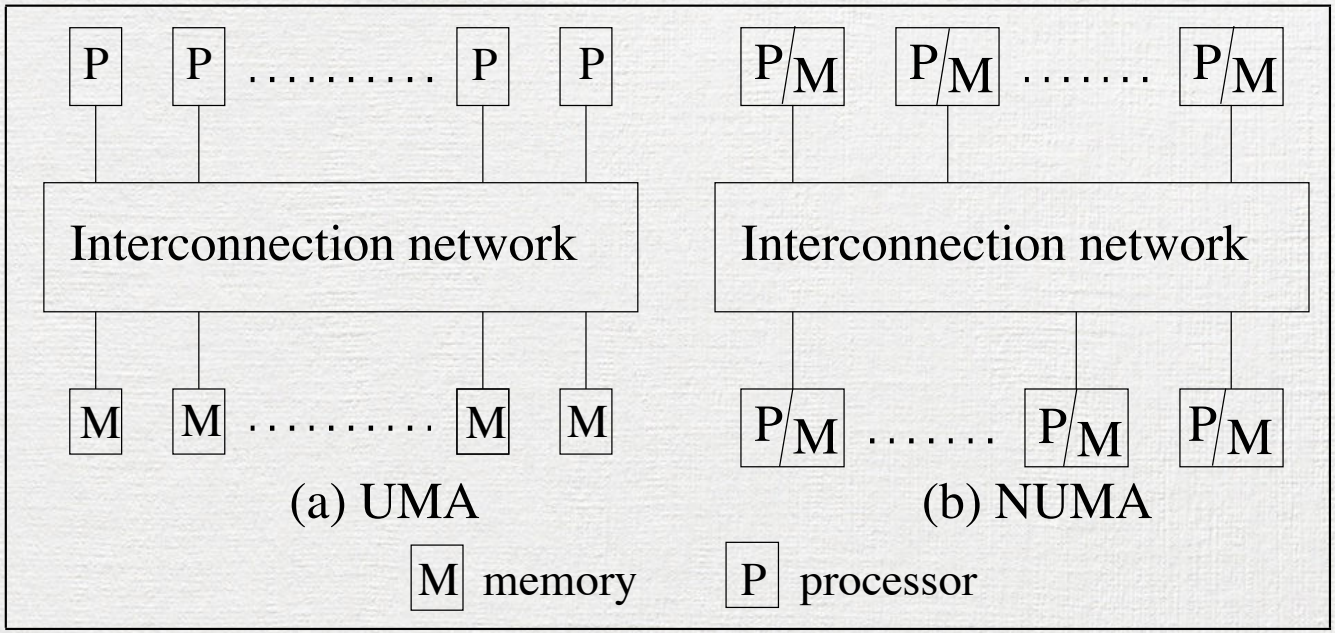
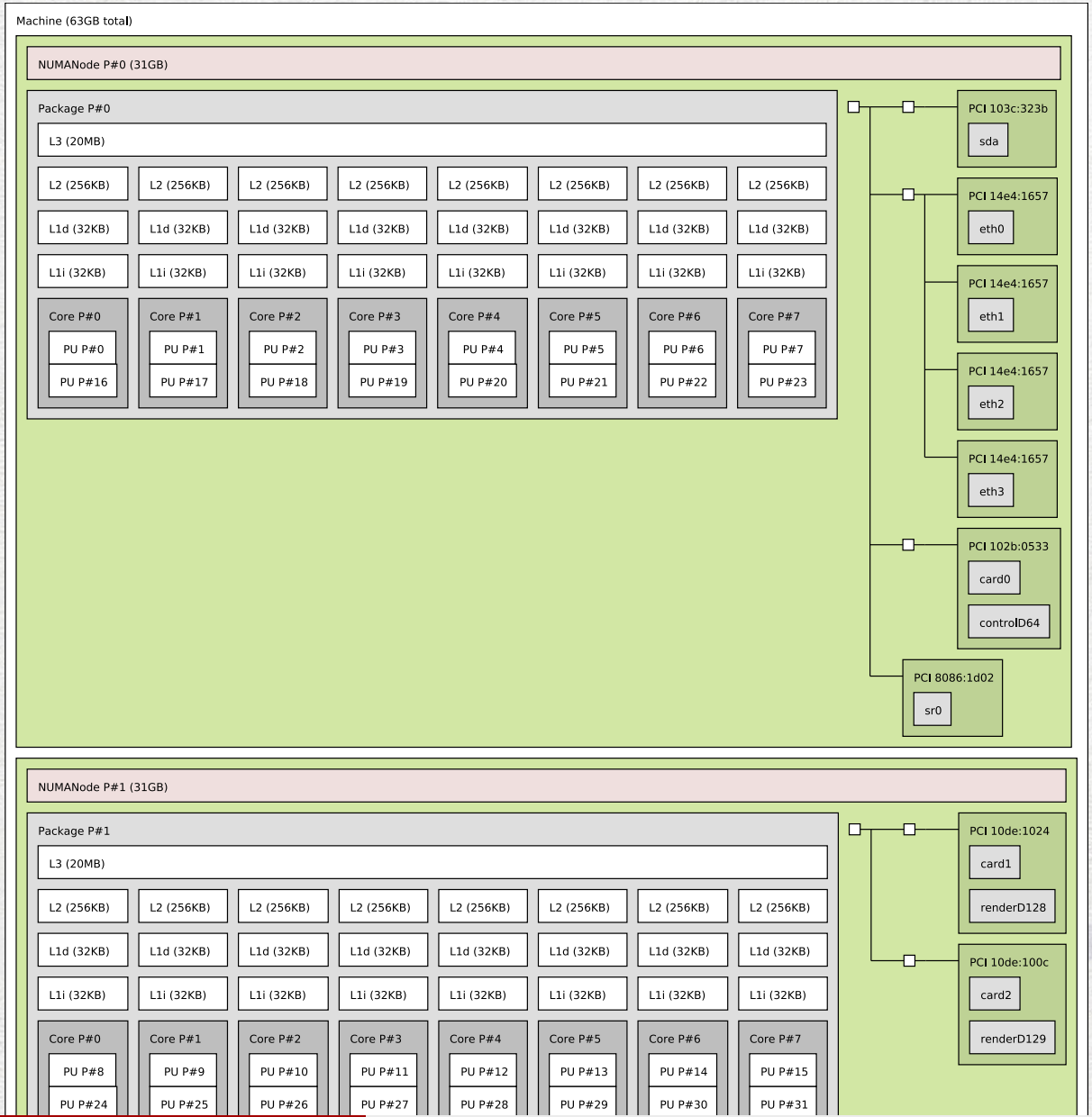
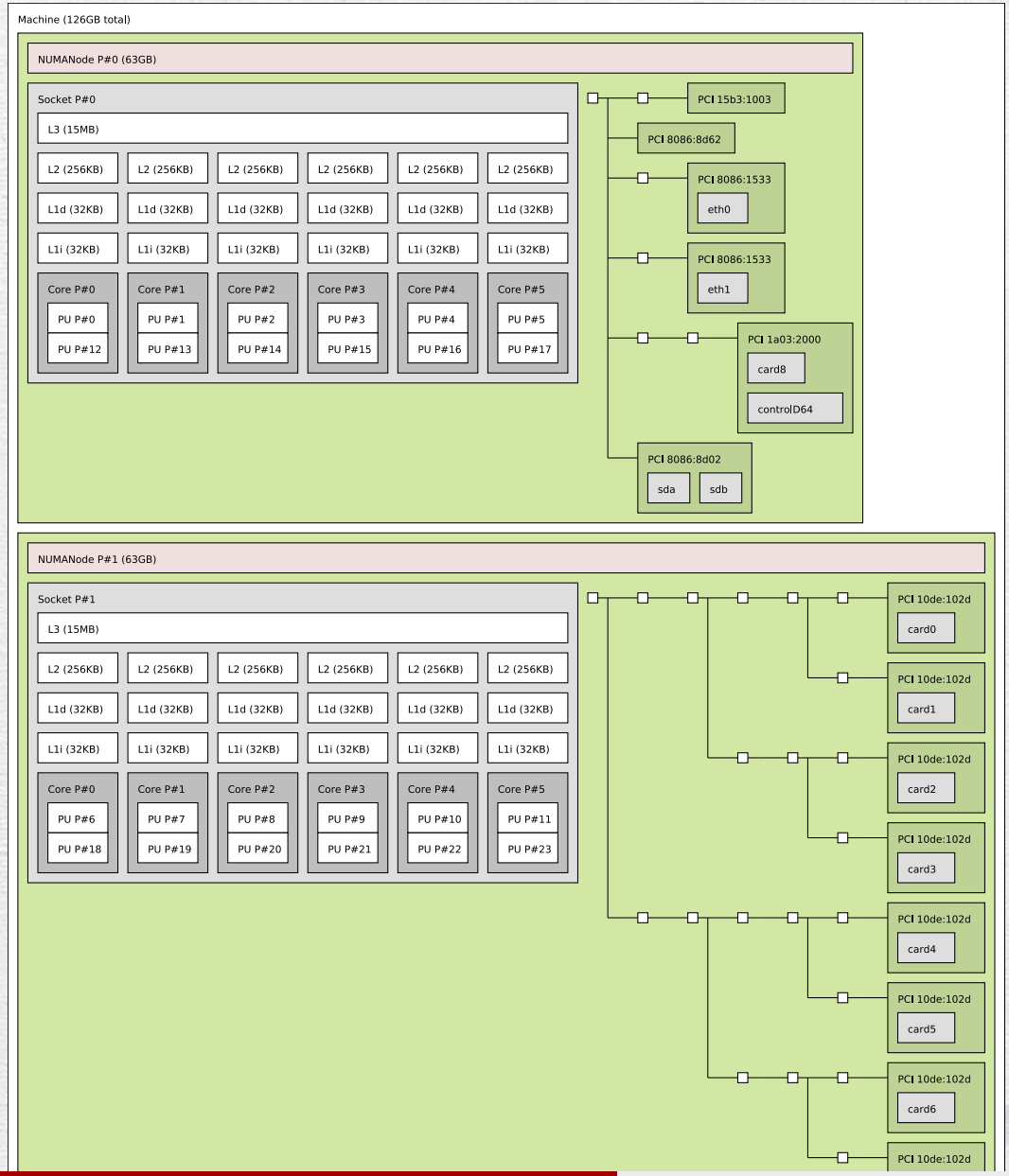


Figure 1: UMA vs NUMA

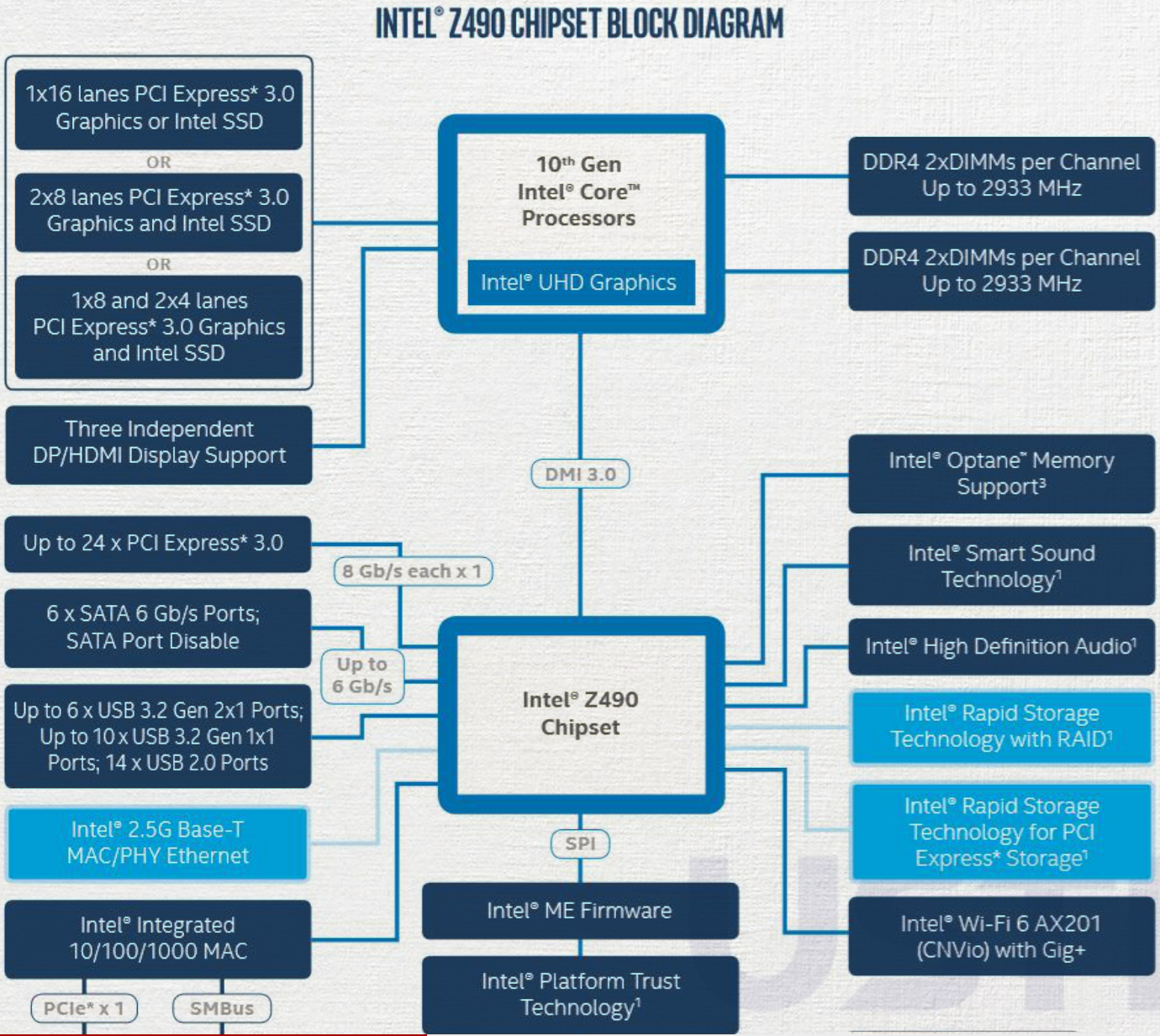
Parallel System: ICTLab's ICT2 NUMA example



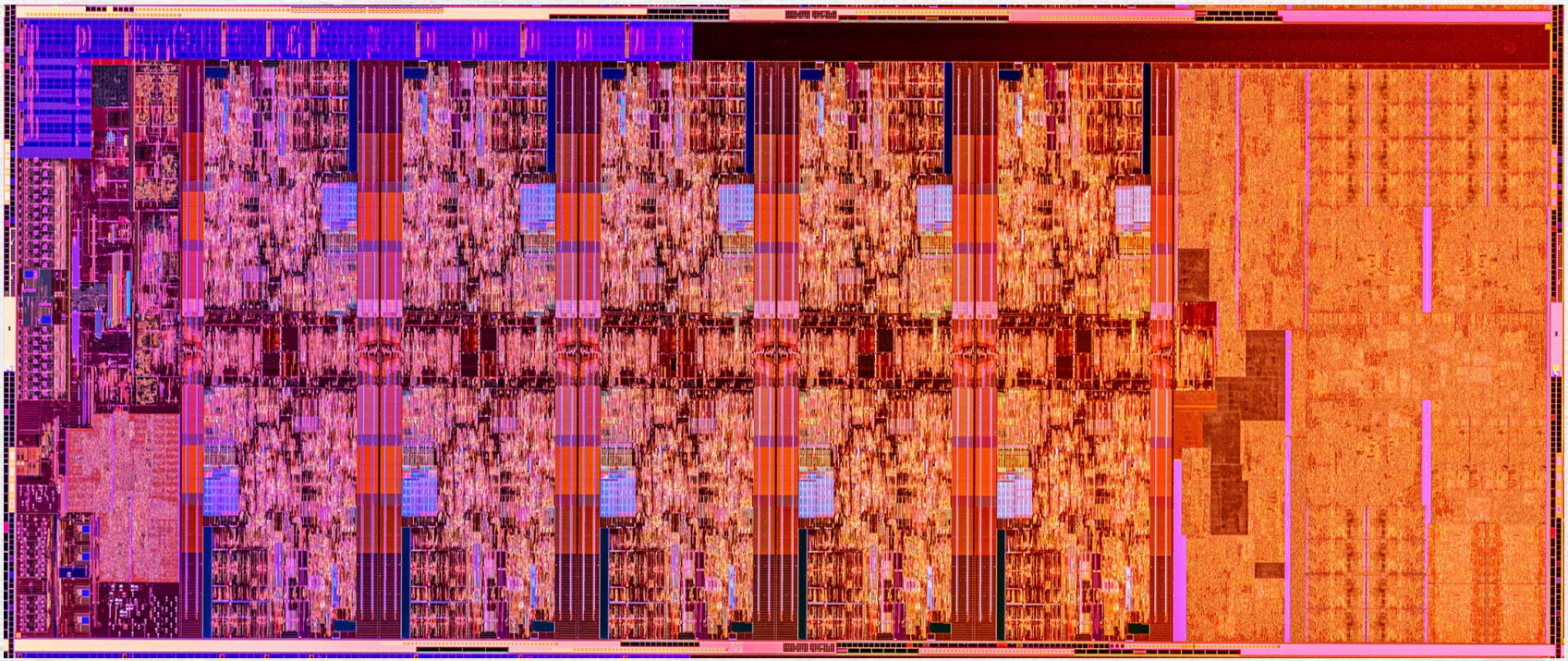
Parallel System: ICTLab's ICT5 NUMA example



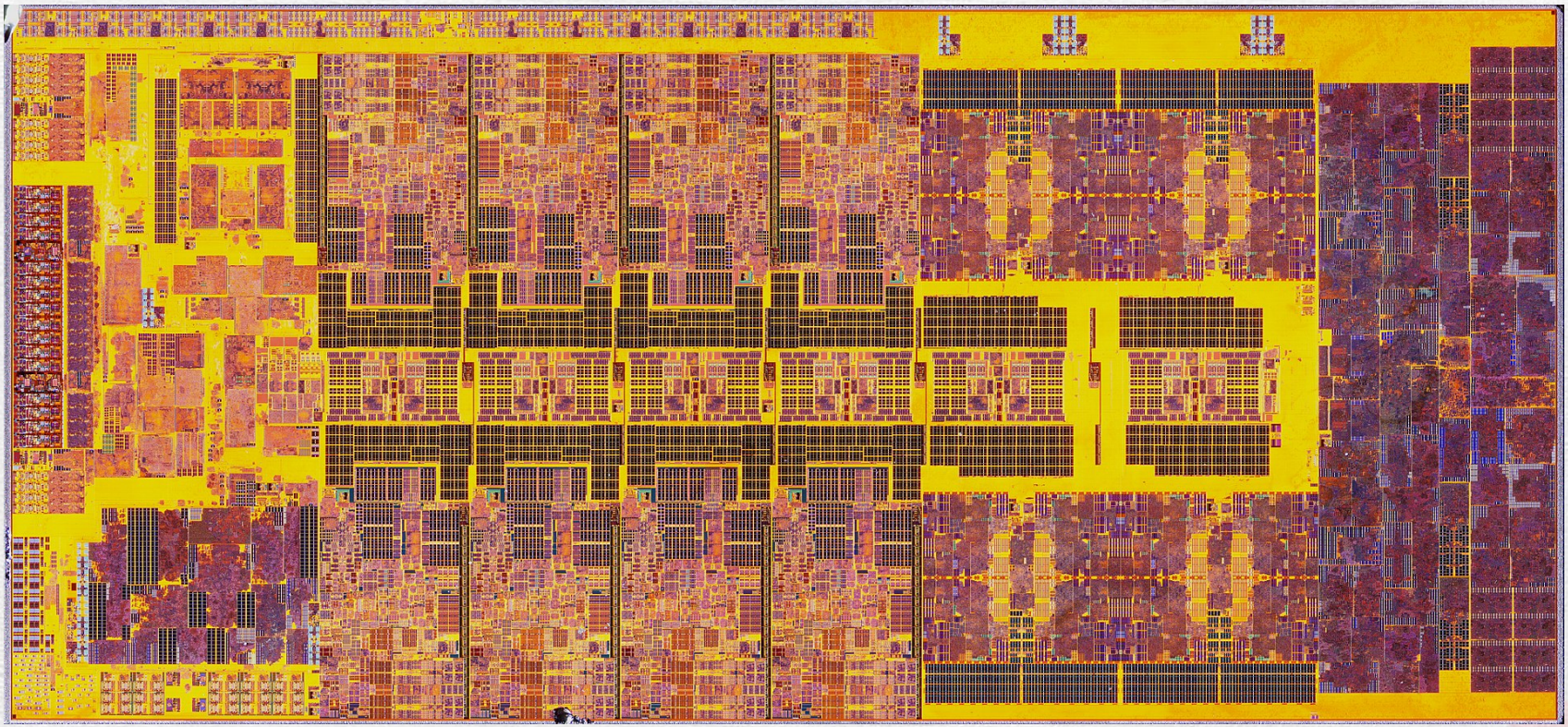
Parallel System: Intel Comet Lake



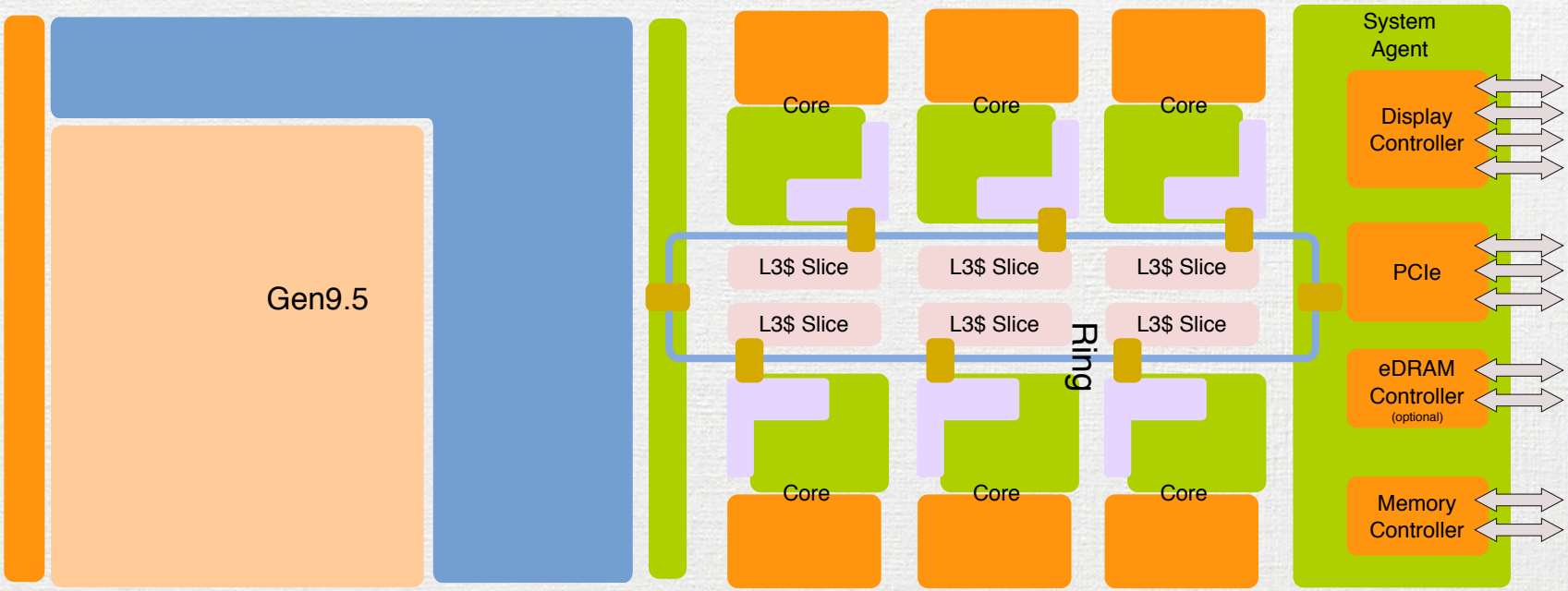
Parallel System: Intel Comet Lake 10850K



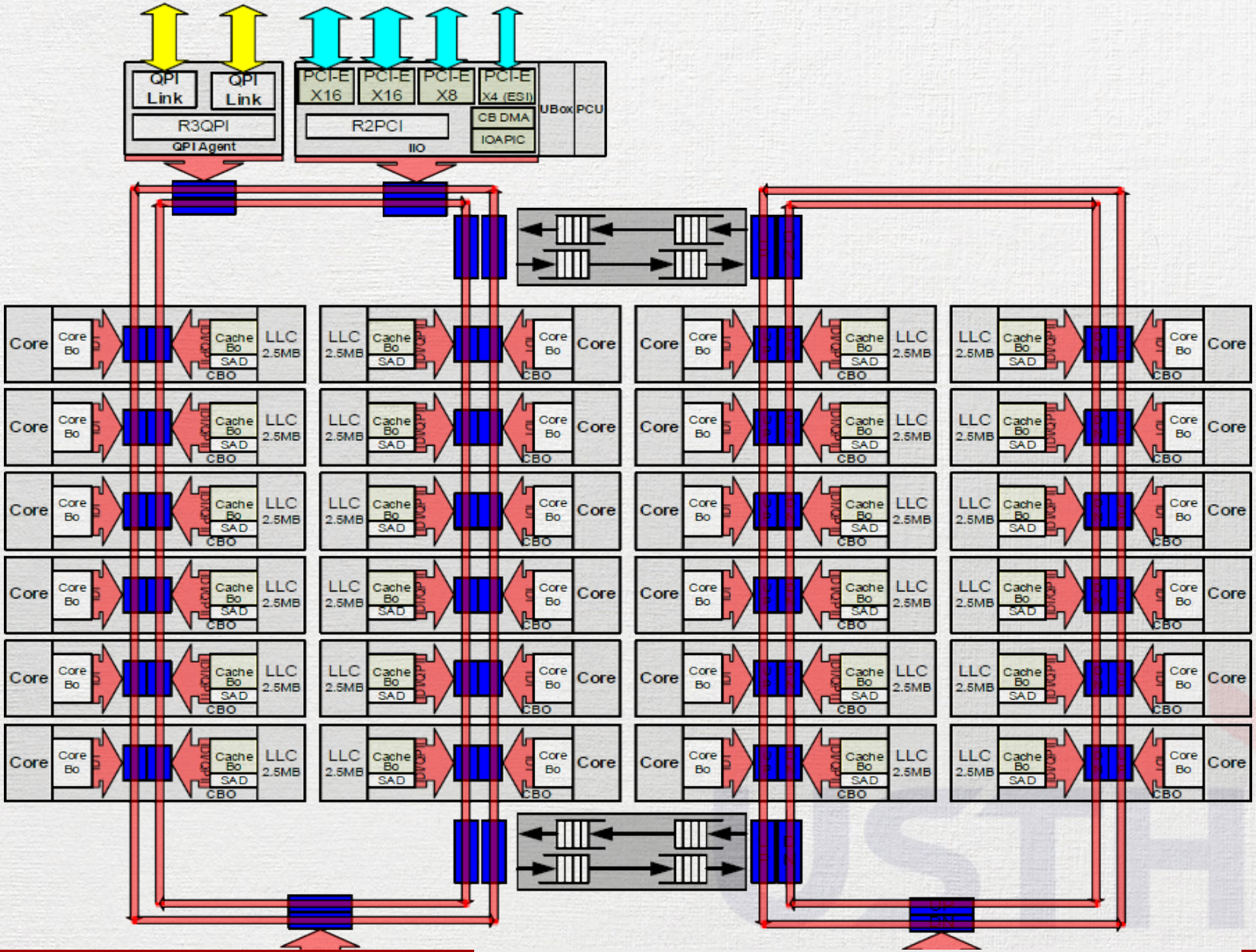
Parallel System: Intel Raptor Lake 13900K



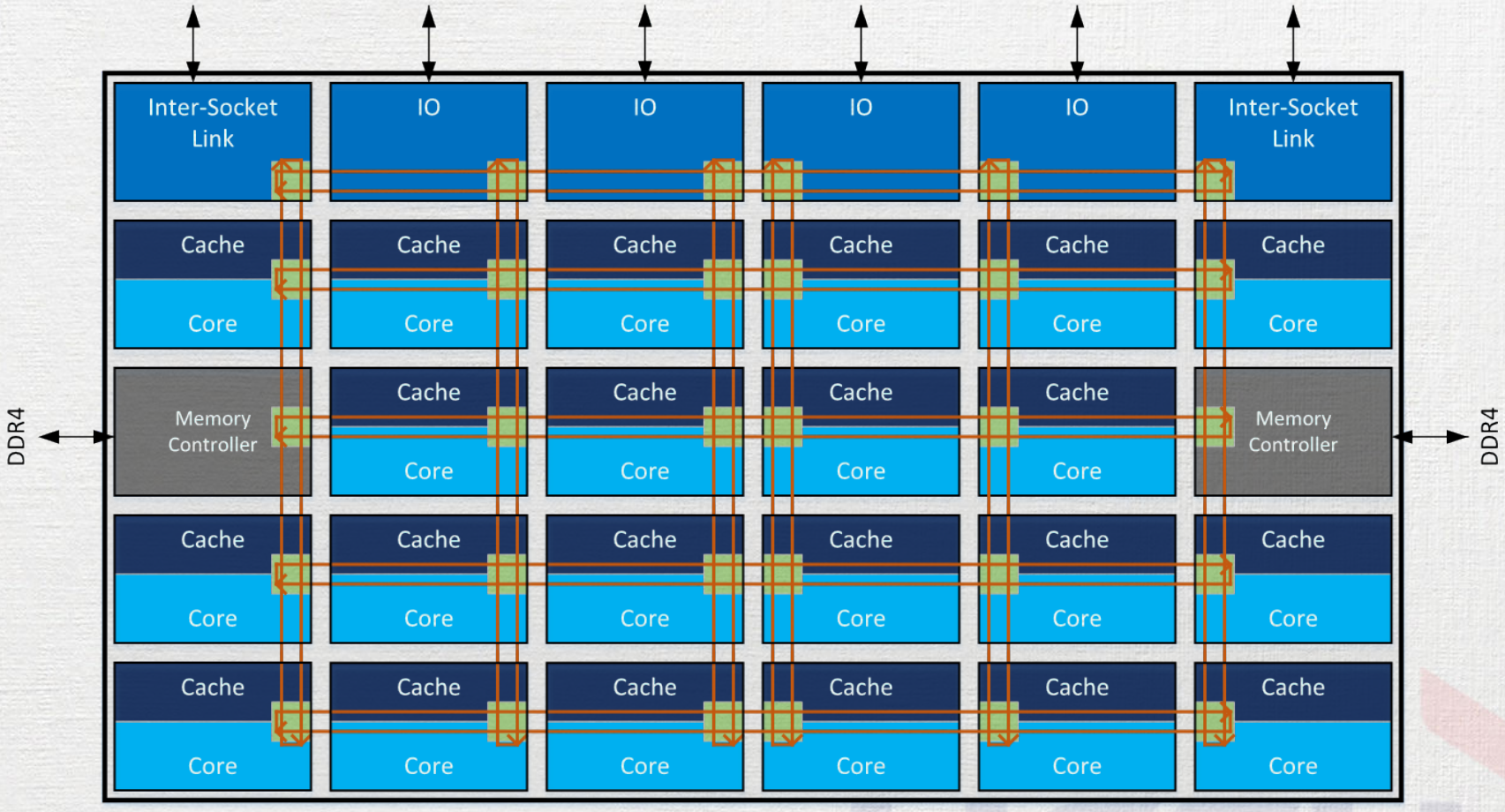
Parallel System: Intel Coffee Lake 8700K



Parallel System: Intel Broadwell EP Xeons



Parallel System: Intel Skylake SP Xeons

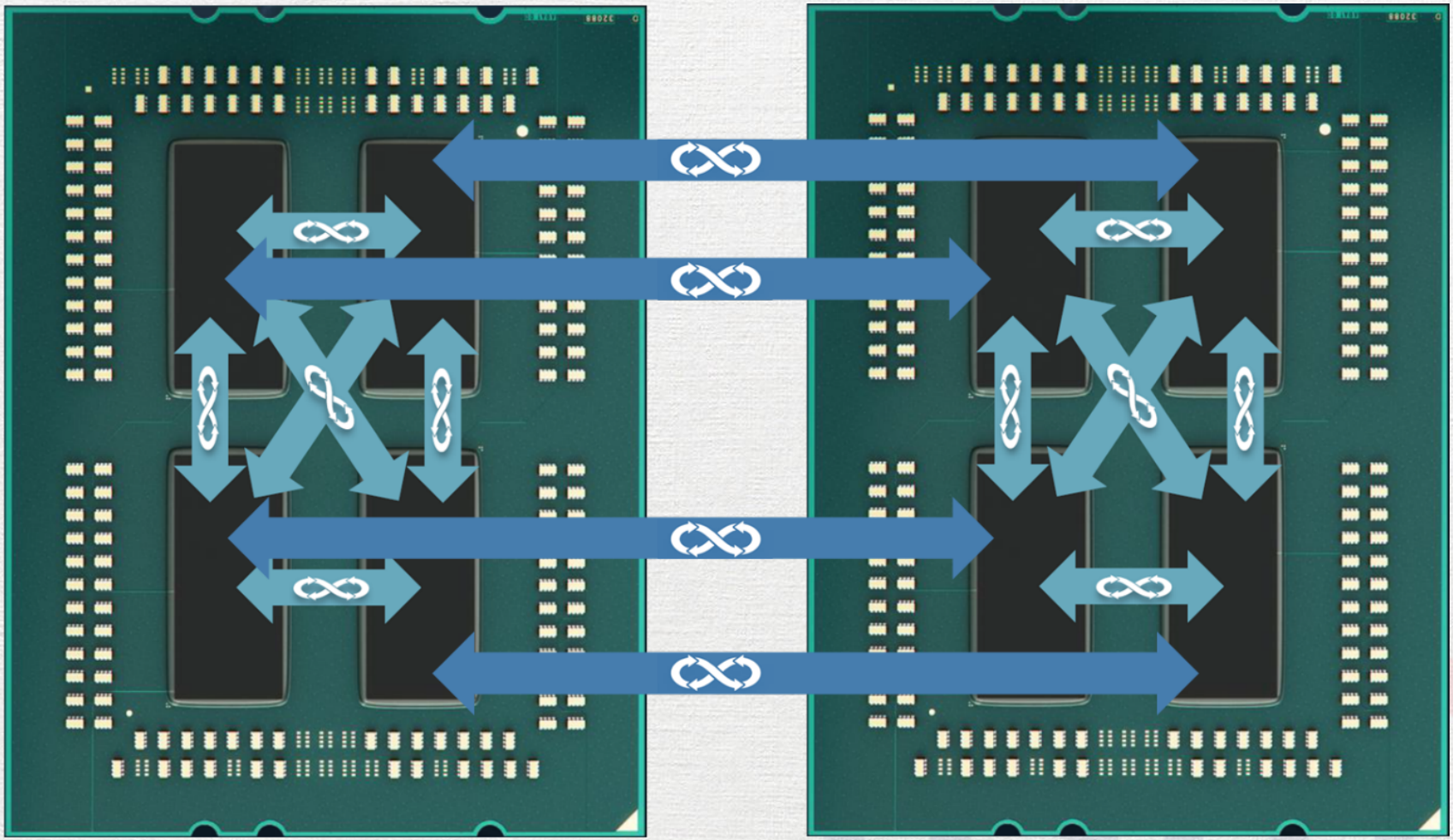


Parallel System: AMD ThreadRipper

- Intra-Socket

Parallel System: AMD Epyc

- Inter-Socket



Why?

- Performance
- Scalability
- Reliability
  - Availability, integrity, fault-tolerance
- Modularity
- Resource sharing
- Performance/cost

Distributed System: Challenges

- Communication
- Processes, scheduling
- Resource naming
- Synchronization
- Storage
- Fault-tolerance
- Security
- Scalability

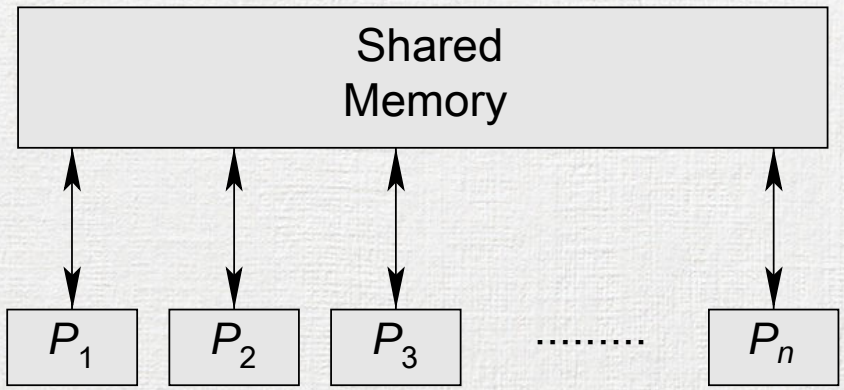
Labwork 0: Hello world!

- Fork course's github repository
  - `https://github.com/SonTG/advancedhpc2022`
- Clone your forked repository
- Update README.md with your name
- Commit and push the change to your forked repository

Parallel Models

PRAM

- Parallel Random Access Machine
- Shared memory
- Multiple processing units



PRAM

Read/write conflicts

- EREW: **E**xclusive **R**ead **E**xclusive **W**rite
- CREW: **C**oncurrent **R**ead **E**xclusive **W**rite
- ERCW: **E**xclusive **R**ead **C**oncurrent **W**rite
- CRCW: **C**oncurrent **R**ead **C**oncurrent **W**rite

Flynn's Taxonomy

- Classification of computer architecture by Michael J. Flynn in 1966
  - SISD: **S**ingle **I**nstruction **S**ingle **D**ata
  - SIMD: **S**ingle **I**nstruction **M**ultiple **D**ata
  - MISD: **M**ultiple **I**nstruction **S**ingle **D**ata
  - MIMD: **M**ultiple **I**nstruction **M**ultiple **D**ata

Flynn's Taxonomy

- Example:

[1] Instruction-level parallelism with pipelining

Flynn's Taxonomy

- Example:
  - SISD: Old school single core (scalar or superscalar[1]) CPUs

[1] Instruction-level parallelism with pipelining

Flynn's Taxonomy

- Example:
  - SISD: Old school single core (scalar or superscalar[1]) CPUs
  - SIMD: GPUs

[1] Instruction-level parallelism with pipelining

Flynn's Taxonomy

- Example:
  - SISD: Old school single core (scalar or superscalar[1]) CPUs
  - SIMD: GPUs
  - MISD: Highly fault tolerance system

[1] Instruction-level parallelism with pipelining

Flynn's Taxonomy

- Example:
  - SISD: Old school single core (scalar or superscalar[1]) CPUs
  - SIMD: GPUs
  - MISD: Highly fault tolerance system
  - MIMD: Modern multi-core CPUs

[1] Instruction-level parallelism with pipelining