BC3.04a Introduction to HPC Programming

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

ICT Department, USTH

BC3.04a Introduction to HPC Programming

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

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

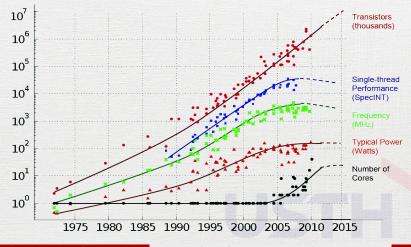
BC3.04a Introduction to HPC Programming

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

Centralized Computing

• One big, fast guy

35 YEARS OF MICROPROCESSOR TREND DATA



BC3.04a Introduction to HPC Programming

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

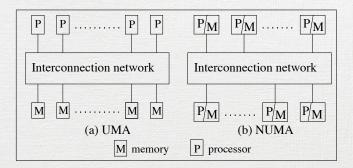


Figure 1: UMA vs NUMA

BC3.04a Introduction to HPC Programming

Parallel Models

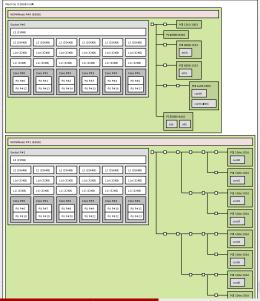
Parallel System: ICTLab's ICT2 NUMA example

Machine (63GB total)	
NURMNode PW0 (31GB)	
Package P#0	PC1103c:323b
L3 (20MB)	sda
L2 (256KB)	PC11464:1657
L1d (32K8)	eth0
L1i (32K8)	PCI1464:1657
Core P#0 Core P#1 Core P#2 Core P#3 Core P#4 Core P#5 Core P#6 Core P#7	eth1
PU P#0 PU P#1 PU P#2 PU P#3 PU P#4 PU P#5 PU P#6 PU P#7 PU P#16 PU P#17 PU P#18 PU P#19 PU P#20 PU P#21 PU P#22 PU P#23	PCI14e4:1657
	eth2
	PCI 1464:1657
	eth3
	PCI 1026:0533
	card0
	controlD64
	PCI 8086:1:002
	se0
NUMANode P#1 (31GB)	
Pactage P#1	PCI 10de:1024
(3 (20140)	card1
L2 (256K8)	renderD128
L1d (32K8)	PCI 10de:100c
L1 (32K8) L1 (32	card2
Core P#0 Core P#1 Core P#2 Core P#3 Core P#4 Core P#5 Core P#5 Core P#5	renderD129
PU P#8 PU P#9 PU P#10 PU P#11 PU P#12 PU P#13 PU P#14 PU P#15 PU P#24 PU P#25 PU P#26 PU P#27 PU P#28 PU P#29 PU P#30 PU P#33	

BC3.04a Introduction to HPC Programming

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

Parallel System: ICTLab's ICT5 NUMA example

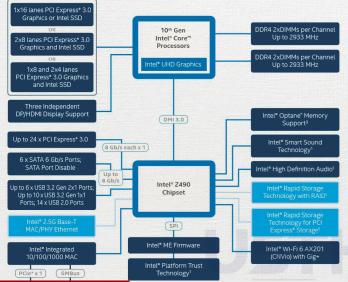


10 / 26

BC3.04a Introduction to HPC Programming

Parallel System: Intel Comet Lake

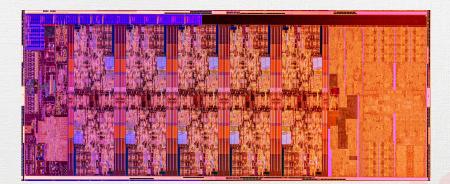
INTEL° Z490 CHIPSET BLOCK DIAGRAM



BC3.04a Introduction to HPC Programming

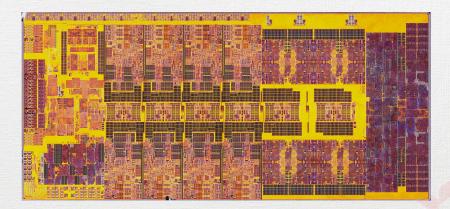
Tran Giang Son, tran-giang.son@usth.edu.vn

Parallel System: Intel Comet Lake 10850K



BC3.04a Introduction to HPC Programming

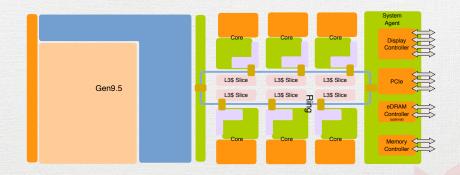
Parallel System: Intel Raptor Lake 13900K



BC3.04a Introduction to HPC Programming

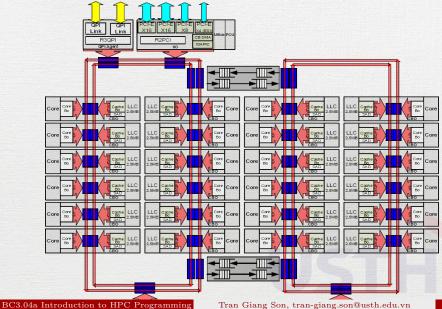
Tran Giang Son, tran-giang.son@usth.edu.vn

Parallel System: Intel Coffee Lake 8700K

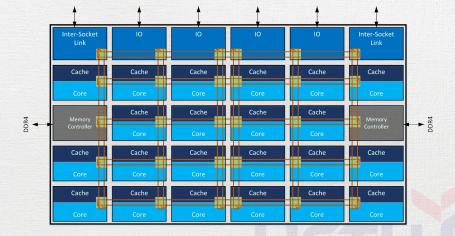


BC3.04a Introduction to HPC Programming

Parallel System: Intel Broadwell EP Xeons



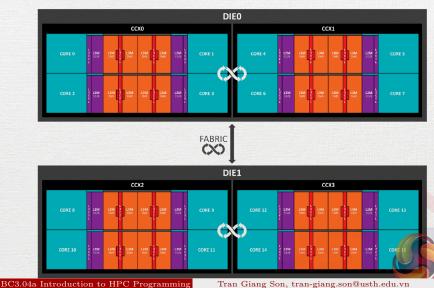
Parallel System: Intel Skylake SP Xeons



BC3.04a Introduction to HPC Programming

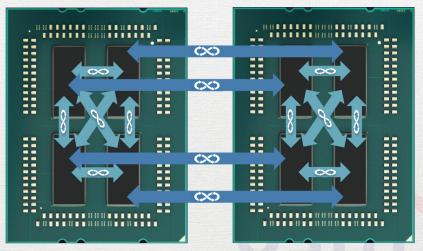
Parallel System: AMD ThreadRipper

• Intra-Socket



Parallel System: AMD Epyc

• Inter-Socket



Why?

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

BC3.04a Introduction to HPC Programming

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

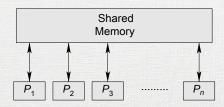
Parallel Models

BC3.04a Introduction to HPC Programming

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

PRAM

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



PRAM

Read/write conflicts

- EREW: **E**xclusive **R**ead **E**xclusive **W**rite
- CREW: Concurrent Read Exclusive Write
- ERCW: Exclusive Read Concurrent Write
- CRCW: Concurrent Read Concurrent Write

- Classification of computer architecture by Michael J. Flynn in 1966
 - SISD: Single Instruction Single Data
 - SIMD: Single Instruction Multiple Data
 - MISD: Multiple Instruction Single Data
 - MIMD: Multiple Instruction Multiple Data

• Example:

¹Instruction-level parallelism with pipelining

BC3.04a Introduction to HPC Programming

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

- Example:
 - SISD: Old school single core (scalar or superscalar¹) CPUs

¹Instruction-level parallelism with pipelining

BC3.04a Introduction to HPC Programming

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

- Example:
 - SISD: Old school single core (scalar or superscalar¹) CPUs
 - SIMD: GPUs

¹Instruction-level parallelism with pipelining

BC3.04a Introduction to HPC Programming

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

- Example:
 - SISD: Old school single core (scalar or superscalar¹) CPUs
 - SIMD: GPUs
 - MISD: Highly fault tolerance system

¹Instruction-level parallelism with pipelining

BC3.04a Introduction to HPC Programming

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

- Example:
 - SISD: Old school single core (scalar or superscalar¹) CPUs
 - SIMD: GPUs
 - MISD: Highly fault tolerance system
 - MIMD: Modern multi-core CPUs

¹Instruction-level parallelism with pipelining BC3.04a Introduction to HPC Programming Tran Giang Son, t

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