CUHK(SZ) Supercomputing Group Introduction

Vito Wu

The Chinese University of Hong Kong, Shenzhen

October 11, 2020

Agenda

- What is Supercomputing Challenge
- Ice Breaking
- How We Conduct Our Research/Study
- Miscellaneous FAQ

What is Supercomputing

- **Definition**
  - A supercomputer is a computer with a high level of performance as compared to a general-purpose computer.
  - The performance of a supercomputer is commonly measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS).
  - Since 2017, there are supercomputers which can perform over 10^{17} FLOPS (a hundred quadrillion FLOPS, 100 petaFLOPS or 100 PFLOPS).

What is Supercomputing

- **Sunway Taihulight** (2016-2018)
  - DEC Alpha 64 Architecture (RISC)
  - 40960 Sunway Processor (260 cores on each die, 10649600 cores in total)
  - R_{Peak}: 125.4359 PFlops
  - R: 93.0146 PFlops

What is Supercomputing

- **Summit** (2018-2020)
  - CPU (POWER9) + GPU (NVLink)
  - 4608 Computation nodes
  - 9216 POWER9 CPU (22 cores on each die)
  - 27648 NVIDIA Tesla V100 GPU
  - *R*: 200 PFlops

What is Supercomputing

- **Fugaku** (2020-)
  - ARM Architecture (A64 SoC)
  - 158976 computation nodes
  - R_{Peak}: 1000 PFlops
  - R: 415 PFlops

What is Supercomputing

- **The Grand Challenges in Computing**
  - Physical Simulations
    - N-body Simulation
    - Weather Forecasting
    - Quantum Simulation
    - Molecular Model
    - ......
  - Machine Learning and Deep Learning Tasks
  - Cryptography and Cryptoanalysis
  - ......
- **Criteria: Performance and Power**

What is Supercomputing

- Main Challenge: How to accomplish computation tasks as fast as possible
  - **Level 1:** Implementation level improvement
    - ▷ Reduce the computation complexity, e.g. $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$
    - ▷ Improve data locality (Register, L1/L2/L3 Cache, Memory, Disk, etc.)
    - ▷ Exploit advanced hardware APIs

What is Supercomputing

- Main Challenge: How to accomplish computation tasks as fast as possible
  - **Level 1:** Implementation level improvement
    - ▷ Reduce the computation complexity, e.g. $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$
    - ▷ Improve data locality (Register, L1/L2/L3 Cache, Memory, Disk, etc.)
    - ▷ Exploit advanced hardware APIs
  - **Level 2:** System level improvement
    - ▷ Compiler (improve how the compiler compiles high level codes into low level)
    - ▷ Operating System (improve how the OS manages memory and processes)
    - ▷ Architectural Design (improve the implementation of ISA)

What is Supercomputing

- Main Challenge: How to accomplish computation tasks as fast as possible
  - **Level 1:** Implementation level improvement
    - ▷ Reduce the computation complexity, e.g. $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$
    - ▷ Improve data locality (Register, L1/L2/L3 Cache, Memory, Disk, etc.)
    - ▷ Exploit advanced hardware APIs
  - **Level 2:** System level improvement
    - ▷ Compiler (improve how the compiler compiles high level codes into low level)
    - ▷ Operating System (improve how the OS manages memory and processes)
    - ▷ Architectural Design (improve the implementation of ISA)
  - **Level 3:** Hardware level improvement
    - ▷ More CPU Cores, more GPU cards, use InfiniBand, etc.
    - ▷ Equip better cooling system
    - ▷ Tuning on BIOS
    - ▷ ......

What is Supercomputing

- Main Challenge: How to accomplish computation tasks as fast as possible
  - **Level 1:** Implementation level improvement
    - ▷ Reduce the computation complexity, e.g. $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$
    - ▷ Improve data locality (Register, L1/L2/L3 Cache, Memory, Disk, etc.)
    - ▷ Exploit advanced hardware APIs
  - **Level 2:** System level improvement
    - ▷ Compiler (improve how the compiler compiles high level codes into low level)
    - ▷ Operating System (improve how the OS manages memory and processes)
    - ▷ Architectural Design (improve the implementation of ISA)
  - **Level 3:** Hardware level improvement
    - ▷ More CPU Cores, more GPU cards, use InfiniBand, etc.
    - ▷ Equip better cooling system
    - ▷ Tuning on BIOS
    - ▷ . . . . . .
  - **Level 4:** Regression (Perf and restart from Level 1)

What is Supercomputing

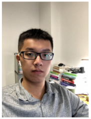
- Rune Knowledge in HPC
  - Being familiar with C/C++ programming(CSC3002)
  - Being familiar with Hardware API / Libraries (Assembly, SIMD, CUDA, . . .)
  - Basic Knowledge on Data Structures and Algorithms(CSC3100)
  - Solid Knowledge on Computer Arch(CSC3050), OS(CSC3150), Principles of Compilation(CSC????)
  - Parallel Algorithms Design(CSC4005)
  - Computer Networks
  - Experience in basic configuration and tools on Linux
  - other domain knowledge . . . . . .

Ice Breaking

- **Team Introduction**
  - Director: Prof. Yeh-Ching CHUNG
  - UG: 3 Senior, 5 Junior, 3 Sophomore (most of us are **CSE** major)

Prof. Yeh-Ching Chung

Dr. Yong-Gang Li

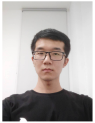


Chen-Hao Wu

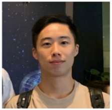
Wei-Dong Gao

Xuan-Chen Wu

Ze-Yin Zhang

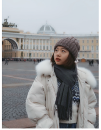


Hao-Tian Xie



Jie-Kun Yang

Su-Pei Yang



Xiao-Nan Qi

- and also many sophomore working in our group...

How We Conduct Our Research/Study

- **Objective:**
  1. Participate in Top-tier Student Cluster Competitions
  2. Study on state-of-the-art HPC knowledge (HW/SW APIs, parallel programming paradigms, optimization techniques, etc.)
  3. Learn how to assemble and configure a cluster system
  4. (extra) Share valuable ideas and conduct research work

How We Conduct Our Research/Study

- **Target Competitions**
  - ASC Student Supercomputer Challenge (East Asia, year-round, 1-5, **has hardware support**)
    `http://www.asc-events.org/`
  - ISC Student Cluster Competition (Europe, year-round, 6-9, **no hardware support**)
    `https://www.isc-hpc.com/student-cluster-competition.html`
  - SC Student Cluster Competition (US, year-round, 11-12, **no hardware support**)
    `https://sc20.supercomputing.org/`
- **General Procedure of Supercomputing Competition**
  1. Setup a cluster (Configure CPU, GPU, memory, disk, network, OS, etc)
  2. Run 2-3 widely-used benchmarks (HPCC, HPCG, HPL, etc)
  3. Run and optimize 2-6 HPC applications
  4. Obtain a score according to given criteria

How We Conduct Our Research/Study

- **Seminars**
  - **Hosts:** Supercomputing Group Members, Researchers in Campus
  - **Time:** Sunday Afternoon (if no other errands)
  - **Topics:**
    1. Tutorial (introduce PL, APIs, etc)
    2. Case Study (on specific problem or the state-of-the-art implementation)
    3. Research Panel (relates to HPC and computer system)
  - **Incoming Seminars**
    1. Use SIMD to Accelerate Programs (10.11)
    2. Optimizing on the Cache Performance (10.18)
    3. GPU Architecture and GPU Programming (10.25)
    4. QUEST Quantum Computer Simulator (11.1)
    5. ......

Misc FAQ

- **What we could offer:**
    - Complementary CS/CE stack knowledge.
      (computer arch, computer network, parallel programming on heterogeneous systems, ...)
    - Architectural thinking to leverage problem solving and system design.
    - Rich opportunity to access state-of-the-art implementations on diverse and active research domains.
      (include but not limit to, mathematics, physics, aerospace engineering, quantum computer science, cryptography and cryptoanalysis, ...)
    - Friends and peers who are able to establish collaborative connections.