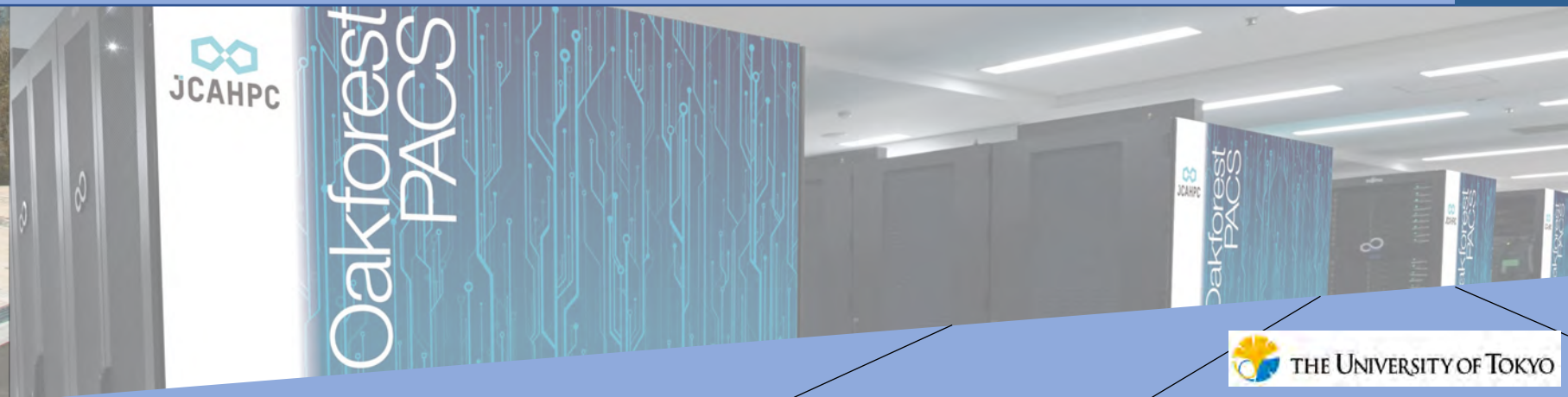
INFORMATION TECHNOLOGY CENTER
THE UNIVERSITY OF TOKYO

JCAHPC

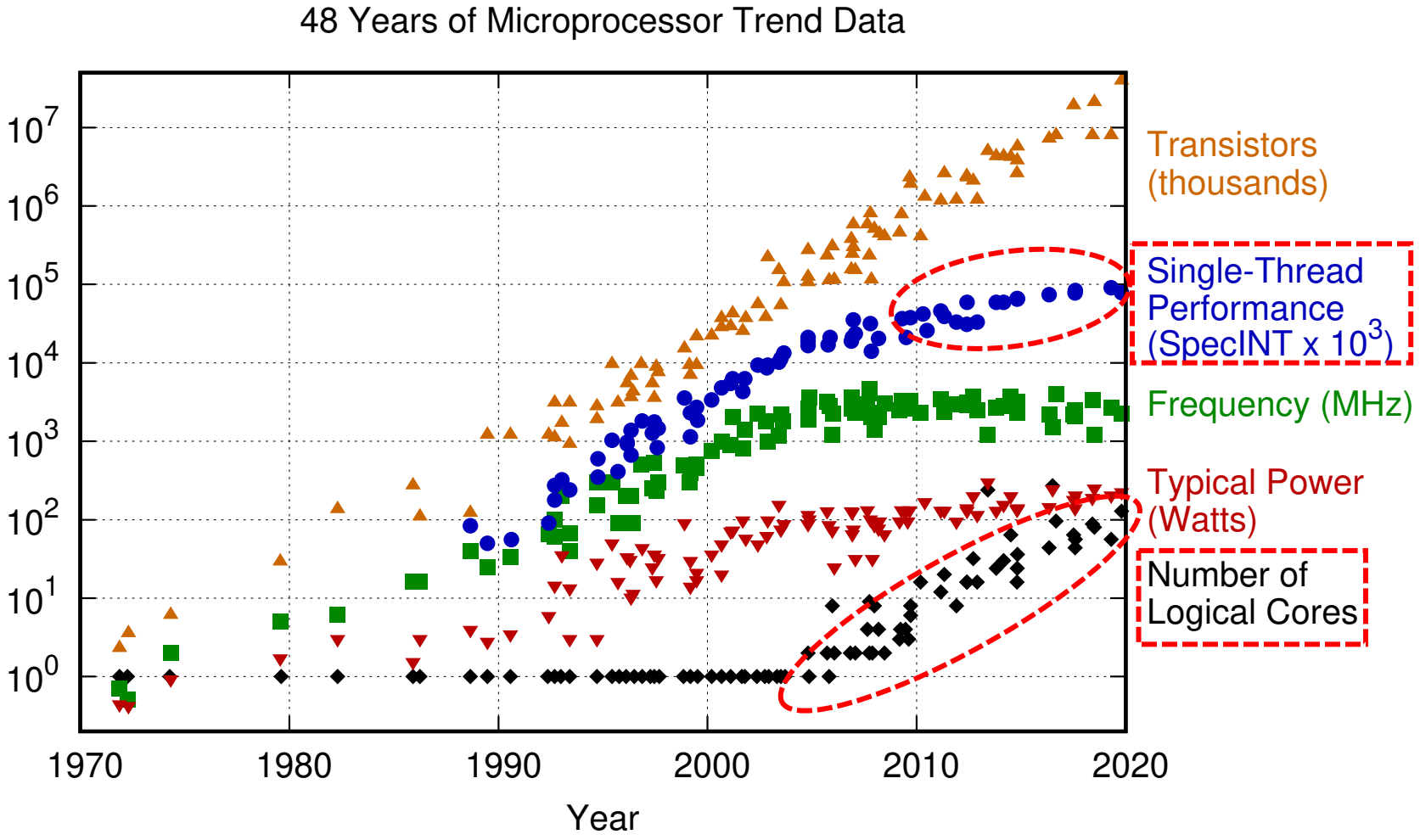
UT-Helper: Support for HPC and DA Utilizing Unused cores

Toshihiro Hanawa



THE UNIVERSITY OF TOKYO

Issues toward Post-moore Era (1/2)



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

- In recent years, single core performance has tended to be constant or rather degraded.
- Increasing the number of cores is support performance improvement
  - Intel Xeon Phi 7250 (KNL, OFP):
    68 core, 272 thread
  - Intel Xeon Platinum 8280 (CascadeLake, OBCX):
    28 core/socket
    56 core/node
  - AMD EPYC Rome:
    64 core, 128 thread/socket,
    128 core, 256 thread/node
  - A64FX (Fugaku):
    48 core +4 or 2 assistant core

Issues toward Post-moore Era (2/2)

Limitation of electricity and heat dissipation

- No longer possible to operate all the CPU cores with full speed simultaneously
  - Power is insufficient
  - Thermal density is too high to remove heat
- Slowdown by all-core operation
  - Ex. Intel AVX512 clock is restricted by occupancy ratio (max. 1GHz down from integer)

Limitation of Memory bandwidth

- Fewer cores may be better result in memory-bound cases
  - Stream benchmark: 32 out of 56 cores is the best on Xeon Platinum 8280 (CascadeLake-EP)

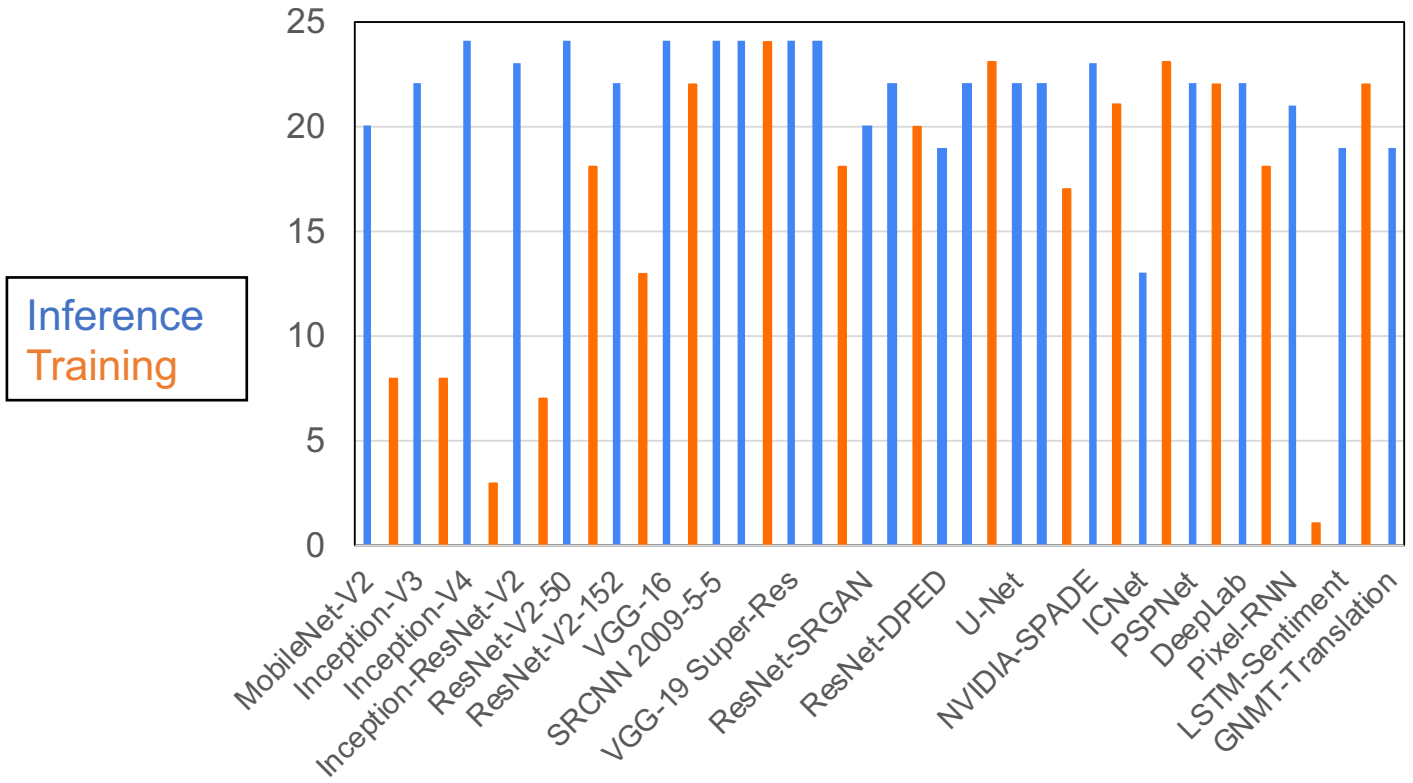
- So far, user should determine the number of the parallelism, but it is complicated to optimize

Control the number of cores for efficient use

- In HPC system, compute node is occupied by single job as usual
- Novice user tends to specify the maximum number of cores without thinking

cf. Optimal number of cores in AI Benchmark

- Best case of results with various number of cores
- Actually, these benchmarks run consecutively
- Varied characteristics from single core to all cores (24)



To achieve best efficiency,
- necessary to change setting on the way
- it is complicated to investigate optimal condition in advance

AI Benchmark
http://ai-benchmark.com/alpha.html

Envieonment:
Intel Xeon Platinum 8260L (CascadeLake)
1 socket, 24 core, 2.4 GHz

Support for HPC and DA Utilizing Unused cores

Cores that do not contribute to performance improvement = Unused core

- Optimal core assignment for each thread in program
  - Automatic core assignment and specification of parallelism
  - Indicate hints to user

What can we do by unused core?

- Effective use under constraint
  - It should not affect main calculation
- Realize within userland
  - No necessary for administrative permission

- Development of framework **UTHelper**
  - Command line tools
  - Libraries

Supported by JSPS Grants-in-Aid for Scientific Research
(A) ”KAKENHI”, 3 years since FY2020
PIs: Toshihiro Hanawa (Lead), Tetsuya Hoshino,
Yohei Miki, Takashi Shimokawabe, Akihiro Ida

Current plan

- In-situ performance profiling
  - No additional codes is necessary for instruments
  - Measure behavior of main calculation and observe impact of helper functions
- Auto-adjust parallelism
- Cache prefetching
- Auto-adjust power budget
- Hiding communication and file IO
  - Apply to numerical library
- On-the-fly analysis during simulation and in-situ visualization
- Language extension using directive
  - Specify prefetch data, policies, QoS

Summary

- Now, we are studying how “unused core” can be utilized !!
  - Before that, optimal core assignment and specification of proper parallelism are prerequisite.
  - First, we are developing the easy to use tool for achieving optimal core setting properly.
- We believe that we still have some room to improve the total performance utilizing unused cores.