Memory-aware Task Mapping for Heterogeneous Multi-Core Systems

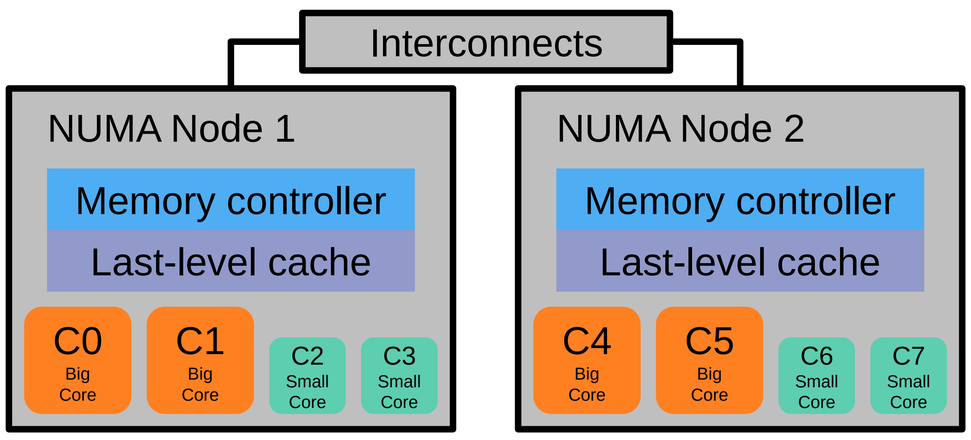
Yifan Jin, Mulya Agung, Jiaheng Liu, Hiroyuki Takizawa

TOHOKU UNIVERSITY

Cyberscience Center

Background

- **Heterogeneous multi-core:** as a new trend in processor design, a multi-core processor has evolved to employ a heterogeneous multi-core architecture integrating some kinds of cores with different performance and energy characteristics on a single chip.
- **NUMA architecture:** the Non-Uniform Memory Access (NUMA) architecture has become a de facto standard in modern HPC systems. This memory architecture brings challenges such as conflicting requirements of minimizing remote access penalty and memory congestion.
- **Task mapping:** determining the allocation of tasks to processor cores, task mapping could significantly affect the usage of systems' heterogeneity and memory resources. A proper task mapping will be a key to high performance and energy efficiency.



NUMA systems built with heterogeneous multi-core processors

Motivation & Objective

Motivation

- Most of the conventional studies on **NUMA awareness** mainly assume to use homogeneous processors and/or processor cores [1].
- Most of the conventional studies on **heterogeneous multi-core** architectures mainly assume the homogeneous interconnection among cores [2].

Objective

- Consider both two factors, **NUMA memory awareness** and **core heterogeneity**, and deal with the task mapping problem on the new system configuration by properly **combining** memory-aware task mapping and heterogeneity-aware task mapping.

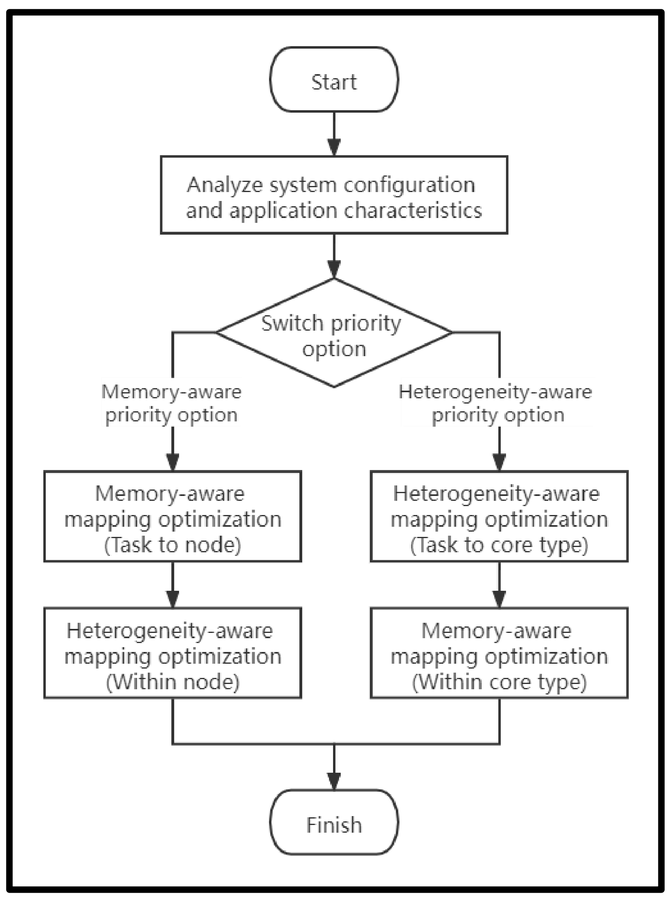
Memory-aware Task Mapping + Heterogeneity-aware Task Mapping

↓

Higher performance and energy efficiency

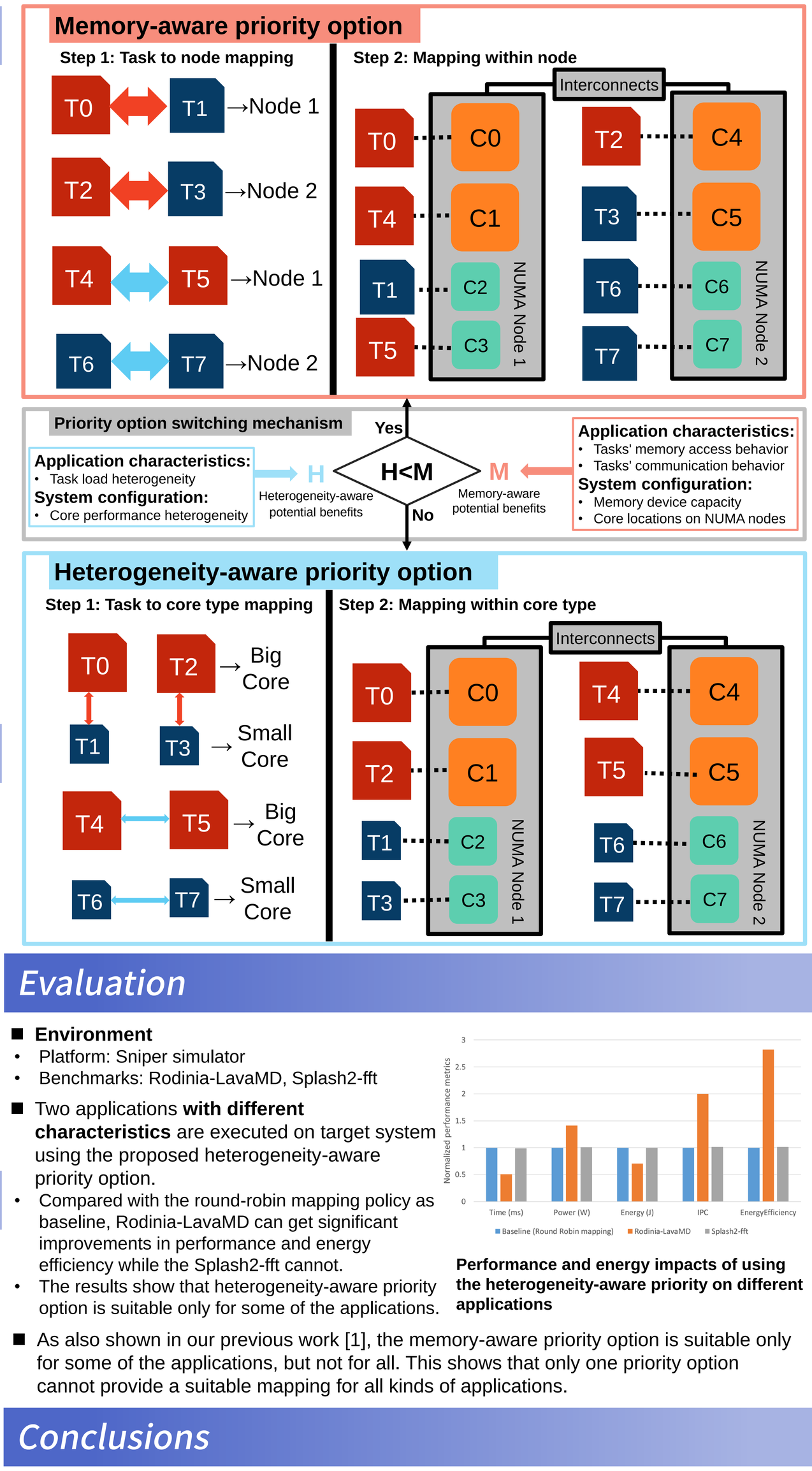
Approach

Task Mapping with Considering both Memory and Heterogeneity



The workflow of the proposed mapping strategy

- Considering the two factors, this work proposes a task mapping strategy that **switches between two priority options** to determine the best mapping for the target application on the target system.
- The two proposed priority options are the **memory-aware priority option** and the **heterogeneity-aware priority option**.
  - Choosing one of the priority options will **prioritize** the impact of that factor at the task mapping.
  - The influence of another factor **will not be ignored but secondary**.
- The proposed priority option switching mechanism **selects appropriate priority options** for individual systems and applications considering their performance characteristics.
- In the following examples, we show application characteristics as follows:
  - Task icon size: task load
  - Arrow width: comm. intensity
  - Same arrow color: concurrent comm.



Evaluation

- **Environment**
  - Platform: Sniper simulator
  - Benchmarks: Rodinia-LavaMD, Splash2-fft
- Two applications **with different characteristics** are executed on target system using the proposed heterogeneity-aware priority option.
  - Compared with the round-robin mapping policy as baseline, Rodinia-LavaMD can get significant improvements in performance and energy efficiency while the Splash2-fft cannot.
  - The results show that heterogeneity-aware priority option is suitable only for some of the applications.

Performance and energy impacts of using the heterogeneity-aware priority on different applications

- As also shown in our previous work [1], the memory-aware priority option is suitable only for some of the applications, but not for all. This shows that only one priority option cannot provide a suitable mapping for all kinds of applications.

Conclusions

- This work focuses on NUMA systems built with heterogeneous multi-core processors and proposes a **new mapping strategy** which includes:
  - **Two priority options:**
    Memory-aware priority option and heterogeneity-aware priority option.
  - **A priority option switching mechanism:**
    Considering the diversity of system configuration and application characteristics.
- Evaluation results on the simulator have shown the necessity of the two mapping priority options and the switching mechanism proposed in this work.
- Our work will also investigate the key performance characteristics of applications that can be used to determine the mapping priority.

References & Acknowledgement

[1] Agung, M., Amrizal, M. A., Egawa, R., & Takizawa, H. (2020). Deloc: A locality and memory-congestion-aware task mapping method for modern numa systems. *IEEE Access*, 8, 6937-6953.
[2] Ding, J. H., Chang, Y. T., Guo, Z. D., Li, K. C., & Chung, Y. C. (2014). An efficient and comprehensive scheduler on Asymmetric Multicore Architecture systems. *Journal of Systems Architecture*, 60(3), 305-314.

This work is partially supported by MEXT Next Generation High-Performance Computing Infrastructures and Applications R&D Program "R&D of A Quantum-Annealing-Assisted Next Generation HPC Infrastructure and its Applications," Grant-in-Aid for Scientific Research(B) #21H03449, and JST SPRING, Grant Number JPMJSP2114.

(URL) https://www.hpc.is.tohoku.ac.jp
(E-mail) jinyifan@hpc.is.tohoku.ac.jp