Enabling Rank- and Iteration-Level Approximate Computing on HPC Applications*†

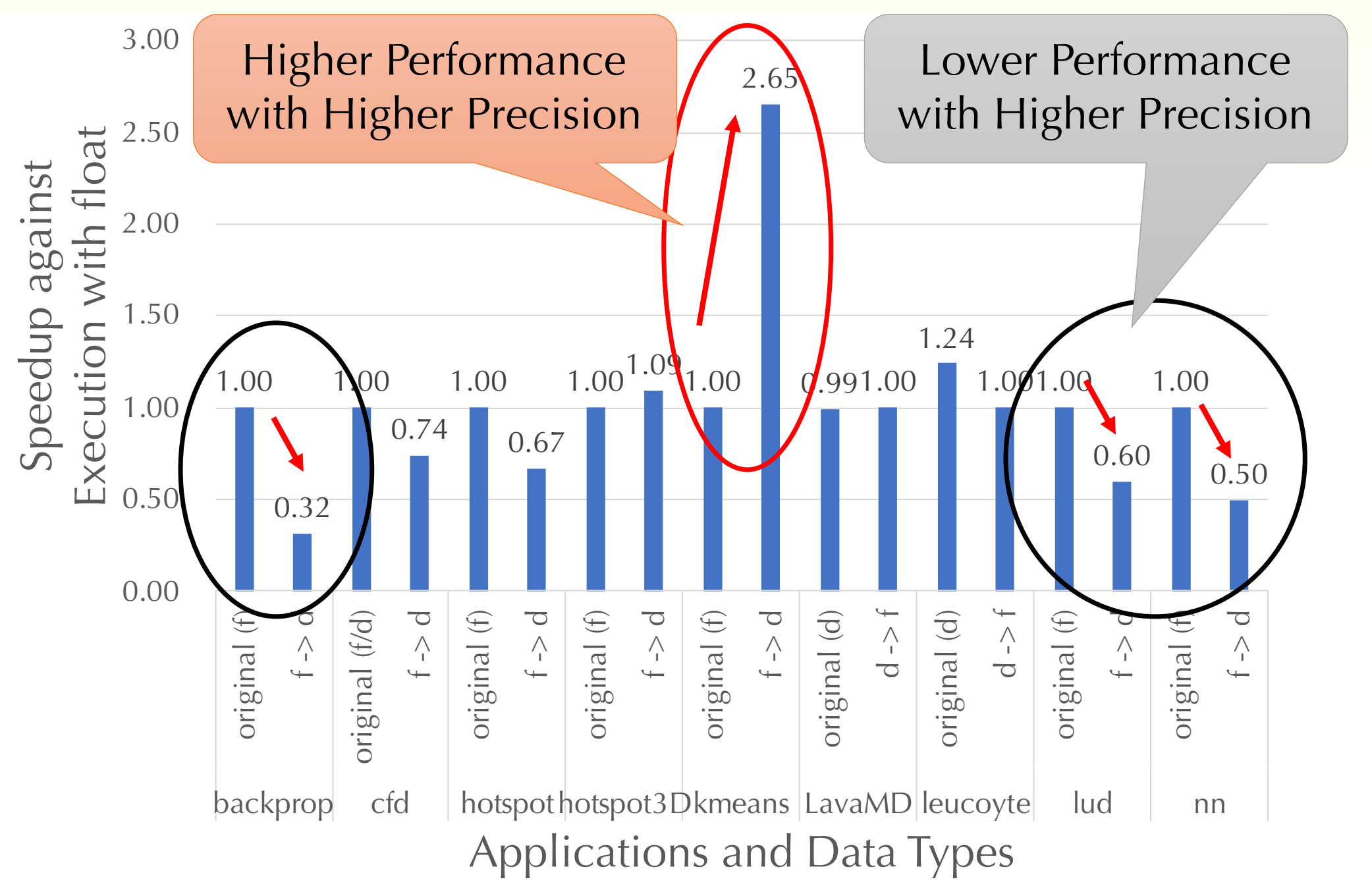
Yasutaka WADA (Meiji Gakuin University), Yoshiyuki MORIE (Teikyo University)
Ryohei KOBAYASHI (Science Tokyo), and Ryuichi SAKAMOTO (Science Tokyo)

†This poster is based on [1] Yasutaka Wada, et al., "Enabling Dynamic Approximate Computing for HPC Applications", Journal of Information Processing, Vol.33, pp.668-674, Oct., 2025.

Approximate Computing (AC) for HPC Systems/Applications

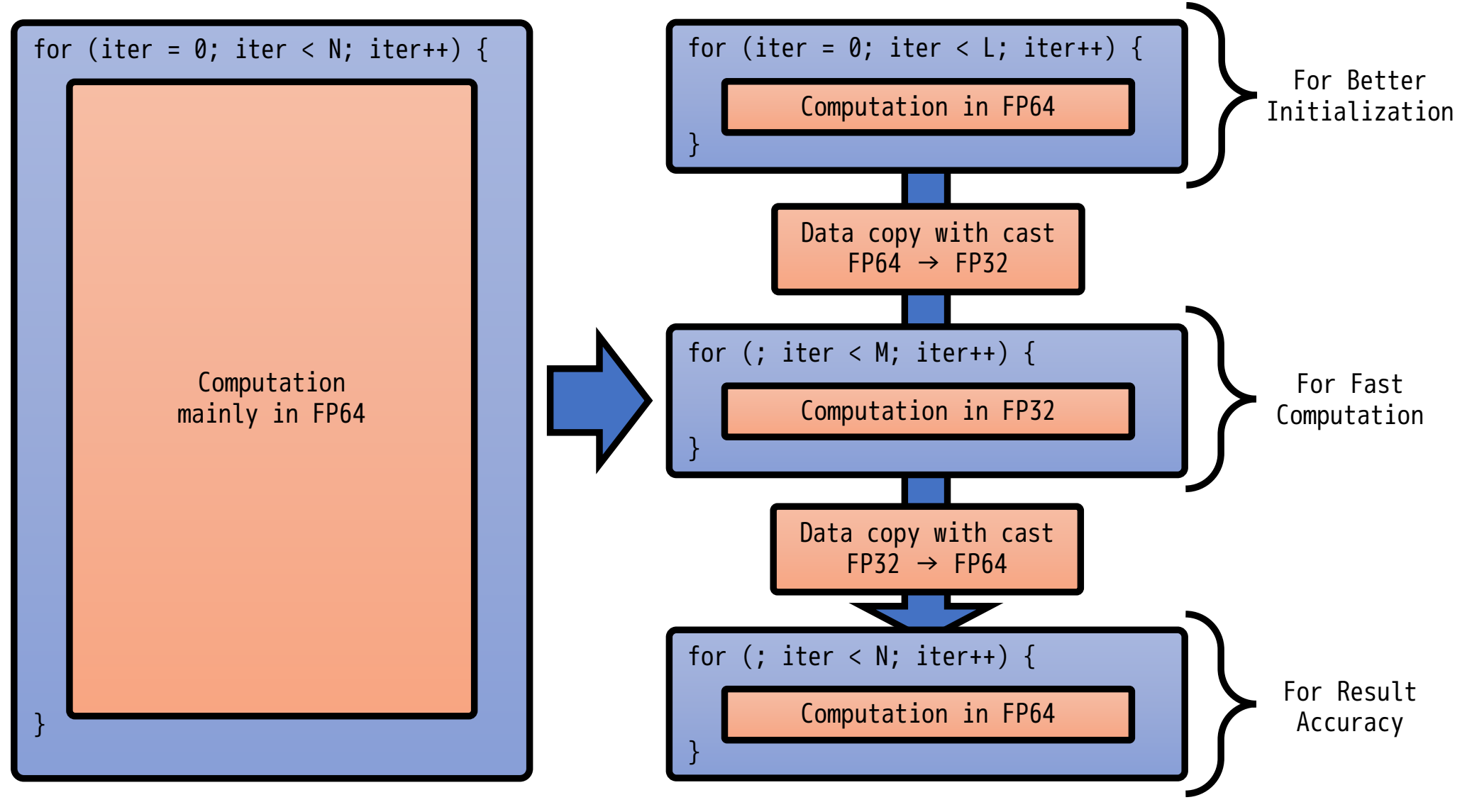
- **Approximate computing optimizes the tradeoff among performance/energy/accuracy**
  - ◆ Effective for applications robust to smaller data precision
    - □ Image processing, Deep learning, etc.
- **Most HPC applications require higher precision to obtain accurate computation results**
  - ◆ Robustness/Sensitivity to data precision depends on their algorithms and structures
    - □ May give better performance with higher precision

→ Needs for Appropriate Approximate Computing for HPC

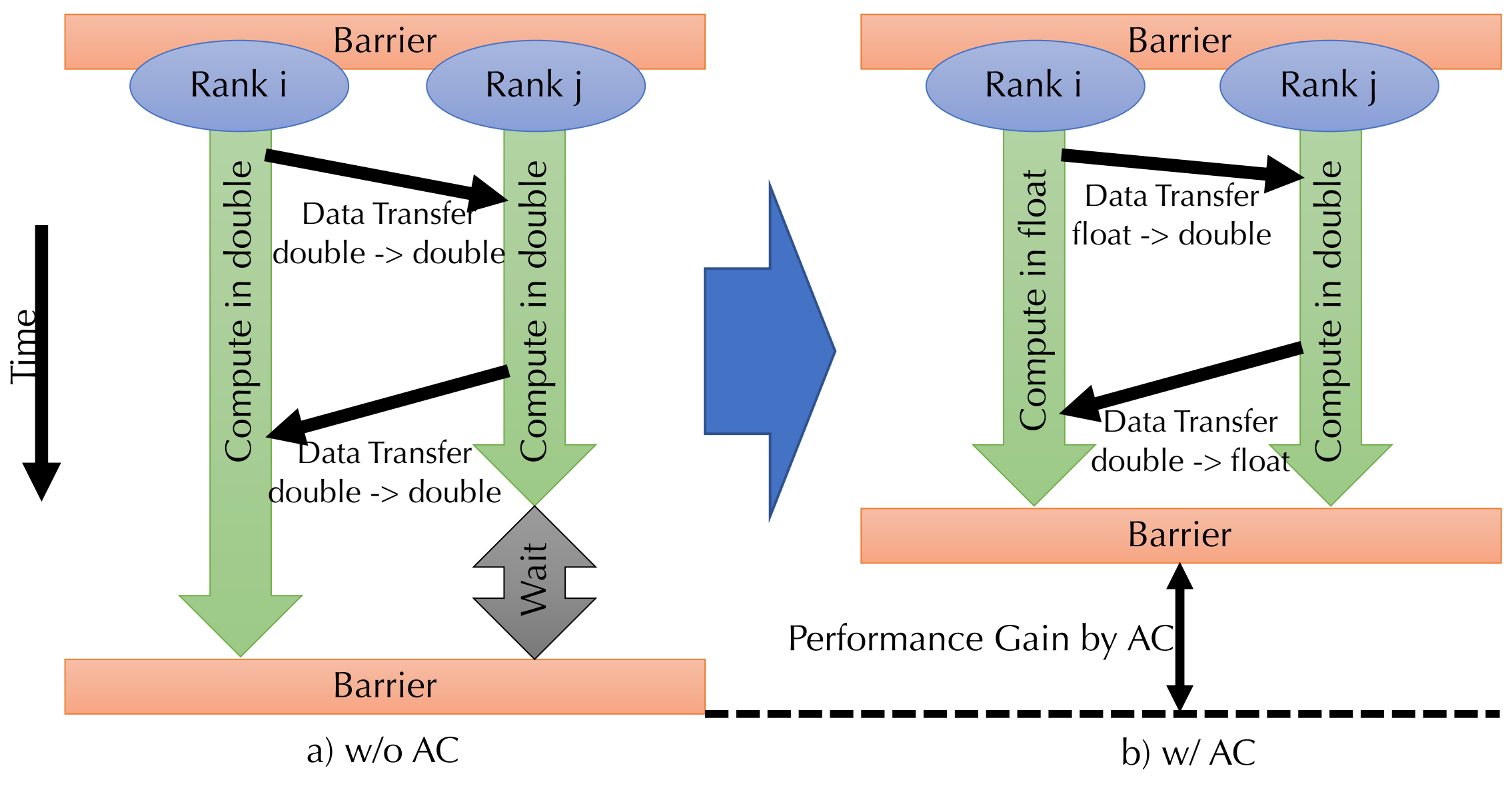


Our Approaches to Realize Dynamic Approximate Computing

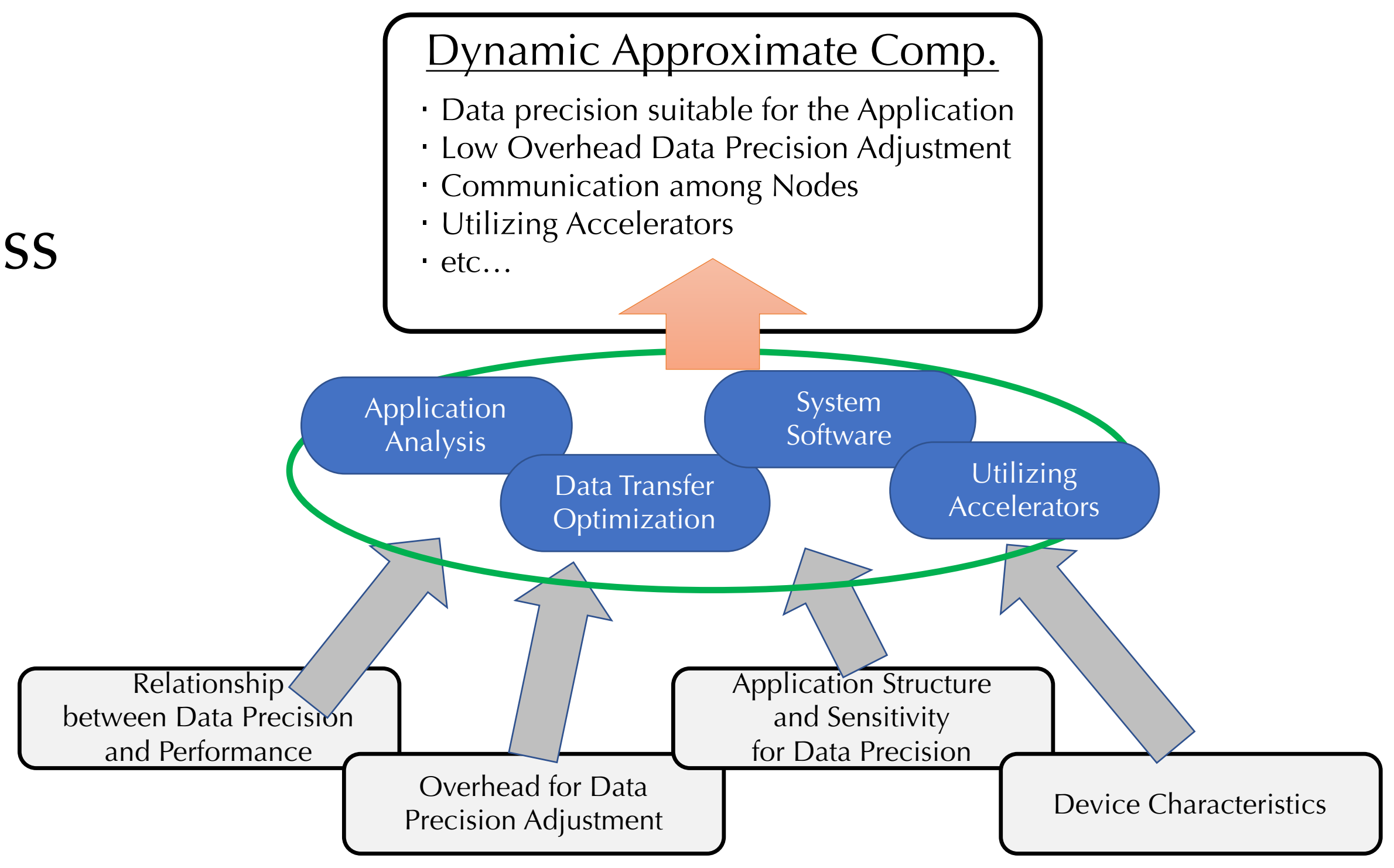
- **Utilize both spatial and temporal structure in HPC applications dynamically**
- **Rank-Level AC:** Each rank can run with its own data precision
- **Iteration-Level AC:** Enable to change data precision according to computation progress


for (iter = 0; iter < N; iter++) {
Computation mainly in FP64
}
for (iter = 0; iter < L; iter++) {
Computation in FP64
}
For Better Initialization
Data copy with cast FP64 → FP32
for (; iter < M; iter++) {
Computation in FP32
}
For Fast Computation
Data copy with cast FP32 → FP64
for (; iter < N; iter++) {
Computation in FP64
}
For Result Accuracy

Iteration-Level AC [1]



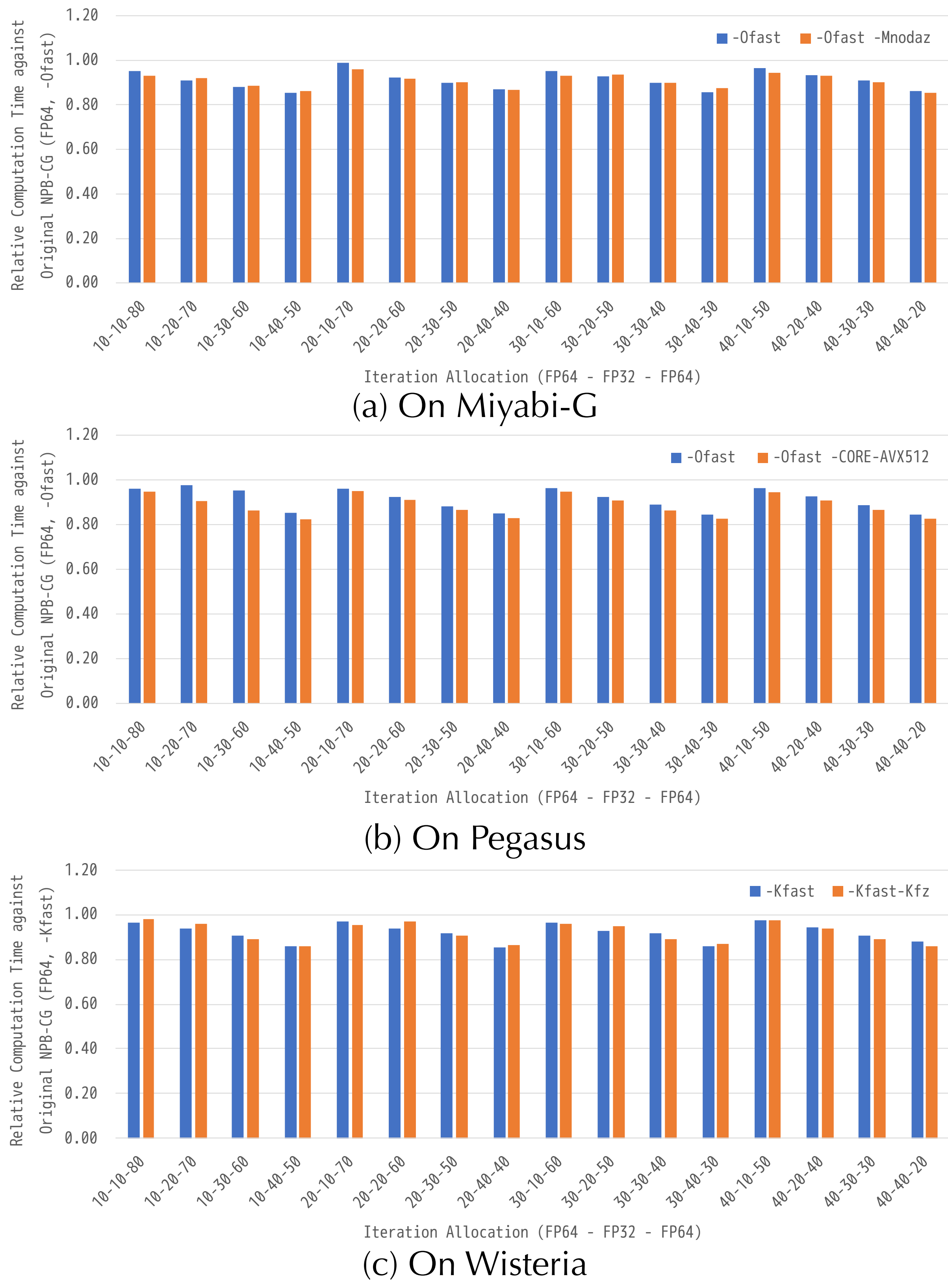
Rank-Level AC [1]


Dynamic Approximate Comp.
・Data precision suitable for the Application
・Low Overhead Data Precision Adjustment
・Communication among Nodes
・Utilizing Accelerators
・etc…
Application Analysis
System Software
Data Transfer Optimization
Utilizing Accelerators
Relationship between Data Precision and Performance
Overhead for Data Precision Adjustment
Application Structure and Sensitivity for Data Precision
Device Characteristics

Required Techniques
to Realize Dynamic AC

Enabling Iteration-Level AC and Communication Overlap for Dynamic AC

- **Iteration-Level AC to utilize temporal characteristics of apps.**
  - ◆ Especially for convergence and time-development loops
  - ◆ Adjust data precision according to the computation progress
    - □ Start with higher precision to stabilize following iterations
    - □ Utilize lower precision to accelerate the execution
    - □ Finalize with higher precision to obtain results accurate enough
  - ◆ Need to keep the results valid for the data type being used
- **Rank-Level AC can be Effective with Communication Overlap**
  - ◆ Changing data precision while operating a data transfer takes much cost
    - □ In terms of both performance and programming/coding
  - ◆ Need simple APIs enable changing data precision within data transfer
    - □ Requires consideration on memory bandwidth
  - ◆ Need to model the data transfer cost with the proposed APIs
    - □ To realize more effective overlap between data transfer and computation

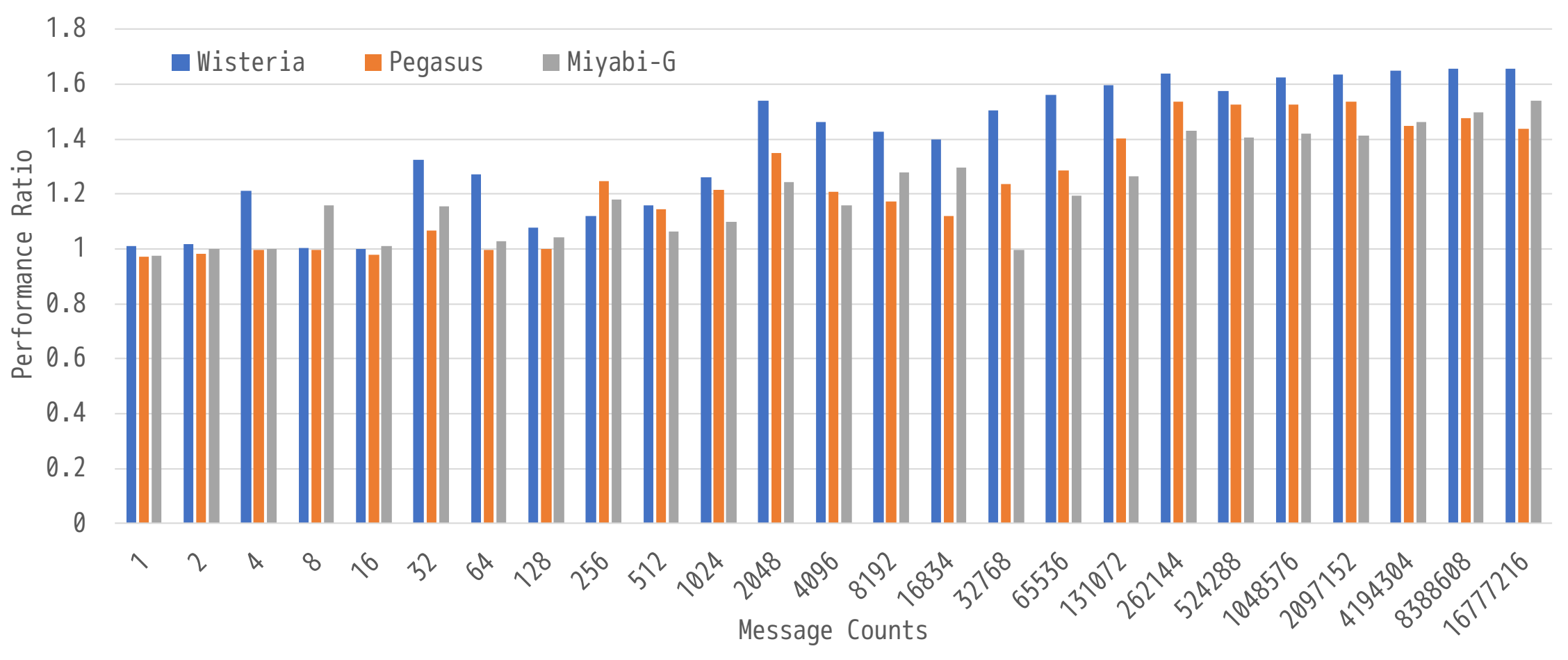


(a) On Miyabi-G

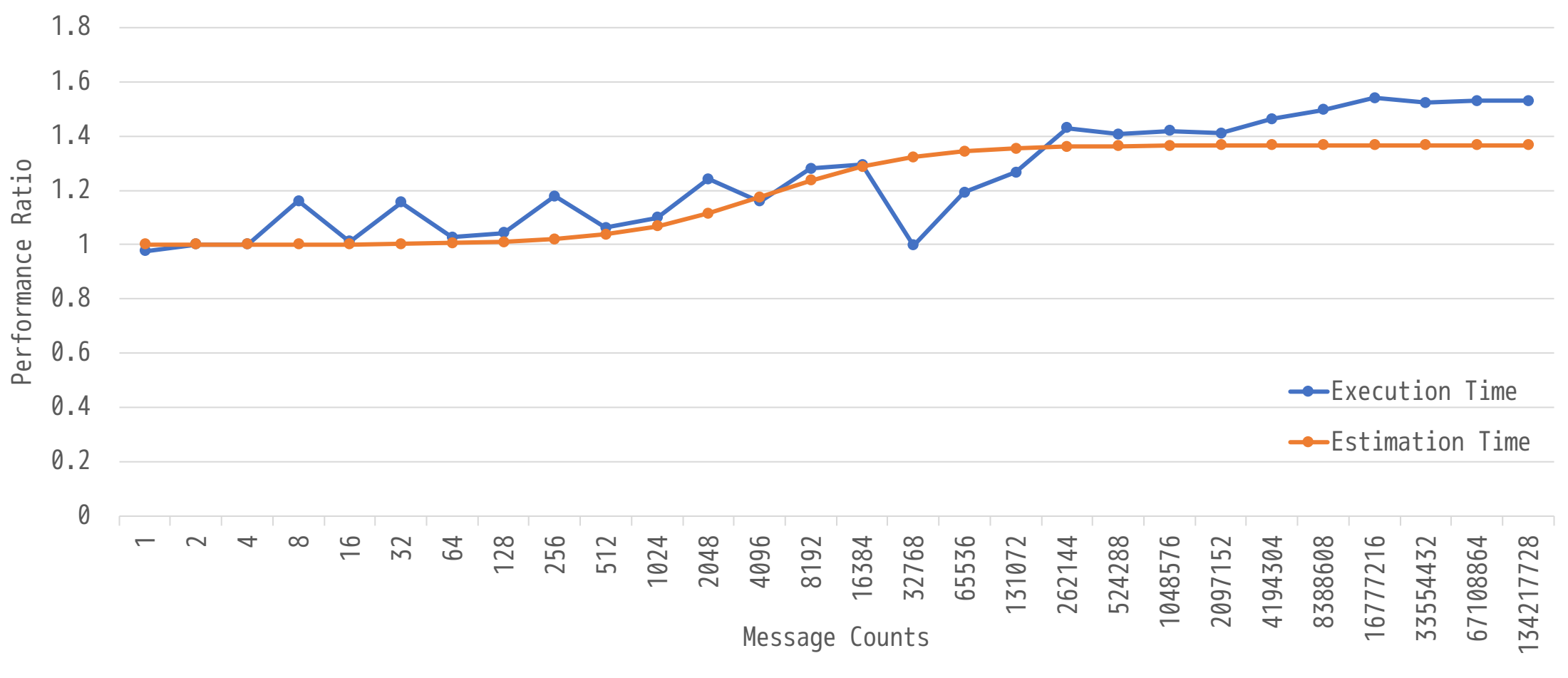
(b) On Pegasus

(c) On Wisteria

Performance Evaluation for Iteration-Level AC
with NPB-CG [1]



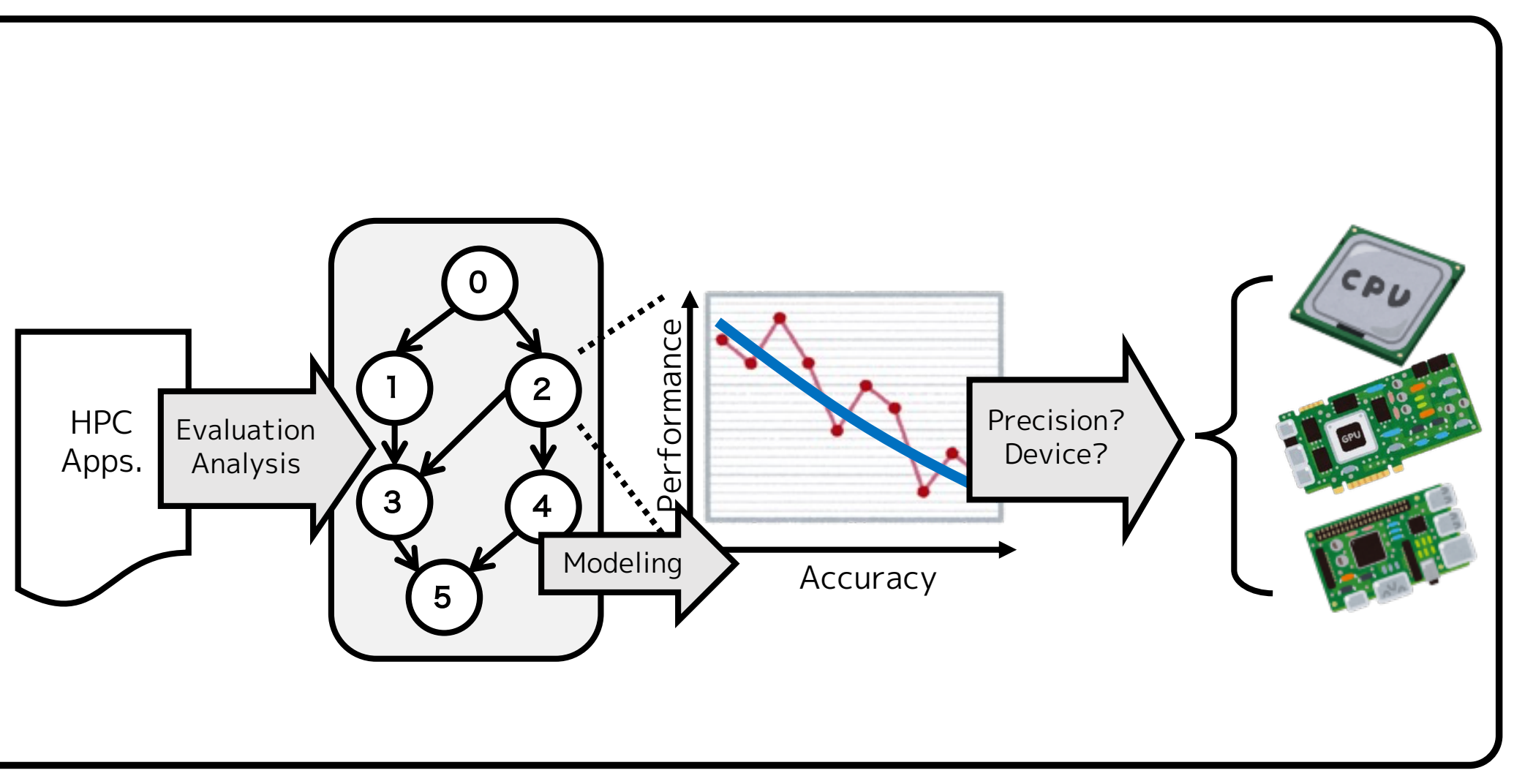
Performance Ratio of the Proposed APIs
against Data Transfer with double/FP64 [1]



Data Transfer Performance on Miyabi-G:
Estimation v.s. Measurement [1]

Future Work for More Effectiveness with Fine-Grained and Dynamic AC

- Need to To utilize compiler techniques to analyze applications and to apply dynamic AC
  - ◆ Provide a performance/accuracy model for each part of the app
  - ◆ (Semi-)automatic application restructuring for iteration- and rank-level AC simultaneously
- Need to adjust data precision automatically and to select appropriate device(s) to be used
  - ◆ Based on the performance models and characteristics of available devices



References:

[1] Yasutaka Wada, Yoshiyuki Morie, Ryohei Kobayashi, Ryuichi Sakamoto, "Enabling Dynamic Approximate Computing for HPC Applications", Journal of Information Processing, Vol.33, pp.668-674, Oct., 2025.
[2] Yasutaka Wada, et al., "Proposal and Preliminary Evaluation of Iteration-Level Approximate Computing Method", IPSJ SIG Technical Report, Vol. 2025-HPC-199, No. 6, pp. 1-5, May, 2025. (in Japanese)
[3] Y. Morie, et al., "Preliminary Evaluation Toward Performance Modeling of High-Throughput Asynchronous Group Communication", IPSJ SIG Technical Report, Vol. 2023-HPC-198, No. 49, pp. 1-6, Mar., 2025. (in Japanese)

*This research was supported in part by Multidisciplinary Cooperative Research Program in CCS, University of Tsukuba, and KAKENHI Grant No. 23K11056 & 25K15141.