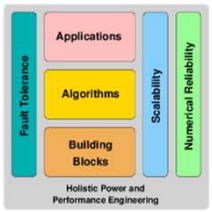
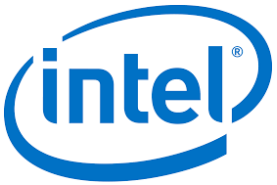
東京大学
THE UNIVERSITY OF TOKYO



東京大学情報基盤センター
INFORMATION TECHNOLOGY CENTER, THE UNIVERSITY OF TOKYO



Parallel Multigrid Methods on Manycore Clusters with IHK/McKernel

Kengo Nakajima[*1,*2], **Balazs Gerofi**[*2], **Yutaka Ishikawa**[*2], **Masashi Horikoshi**[*3]

***1: Information Technology Center, University of Tokyo, *2: RIKEN R-CCS, *3: Intel**

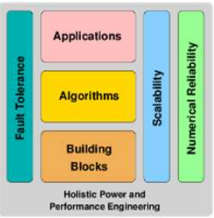
ScalA19: 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems in conjunction with SC19, November 18, 2019, Denver, CO

Acknowledgements

- JST/CREST
- DFG/SPPEXA

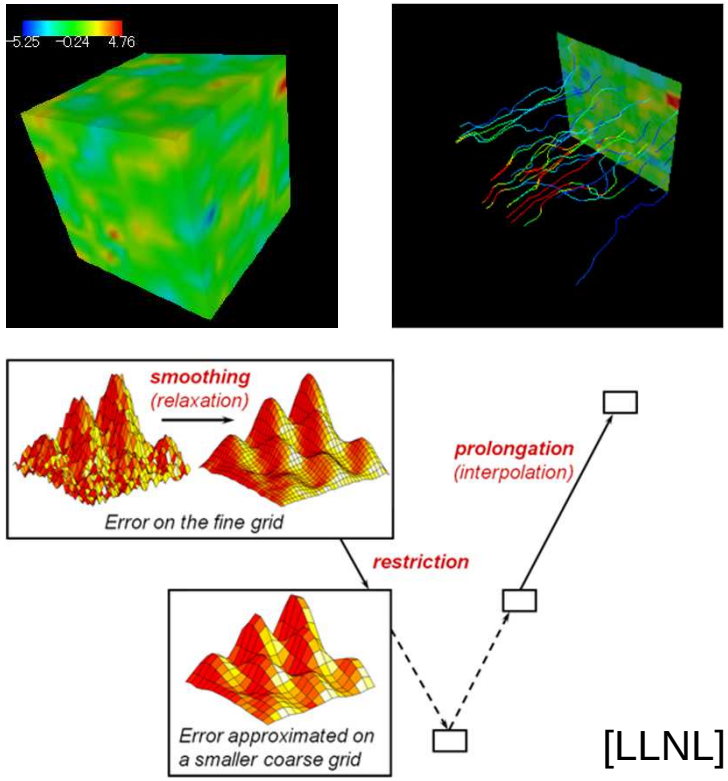
- **JHPCN (jh180022-NAHI, jh180041-NAHI)**
  - **Innovative Multigrid Methods**
- JCAHPC
  - Large-Scale HPC Challenge on Oakforest-PACS

- **Yoshio Sakaguchi (Fujitsu)**



Target Application & HW

- 3D Groundwater Flow via Heterogenous Porous Media: **pGW3D-FVM**
  - Poisson's Eq. (λ=10^{-5}-10^{+5}) $\nabla \cdot (\lambda(x,y,z)\nabla\phi) = q$
  - Finite Volume Method (FVM), Structured Mesh
  - **Conjugate Gradient preconditioned by Multigrid (MGCG), Geometric MG, IC(0) Smoother**
  - Sliced ELL for Storage of Sparse Matrices
- **Multigrid**
  - **Scalable O(N) algorithm, but many problems towards Exascale Computing**
- Oakforest-PACS (OFP)
  - JCAHPC by U.Tsukuba & U.Tokyo, Fujitsu
  - 8,208 Intel Xeon/Phi (Knights Landing (KNL)) CPU's
  - Omni Path Architecture (OPA)
  - 15th in TOP 500 (November 2019)



[LLNL]

JCAHPC

東京大学 THE UNIVERSITY OF TOKYO

筑波大学 University of Tsukuba

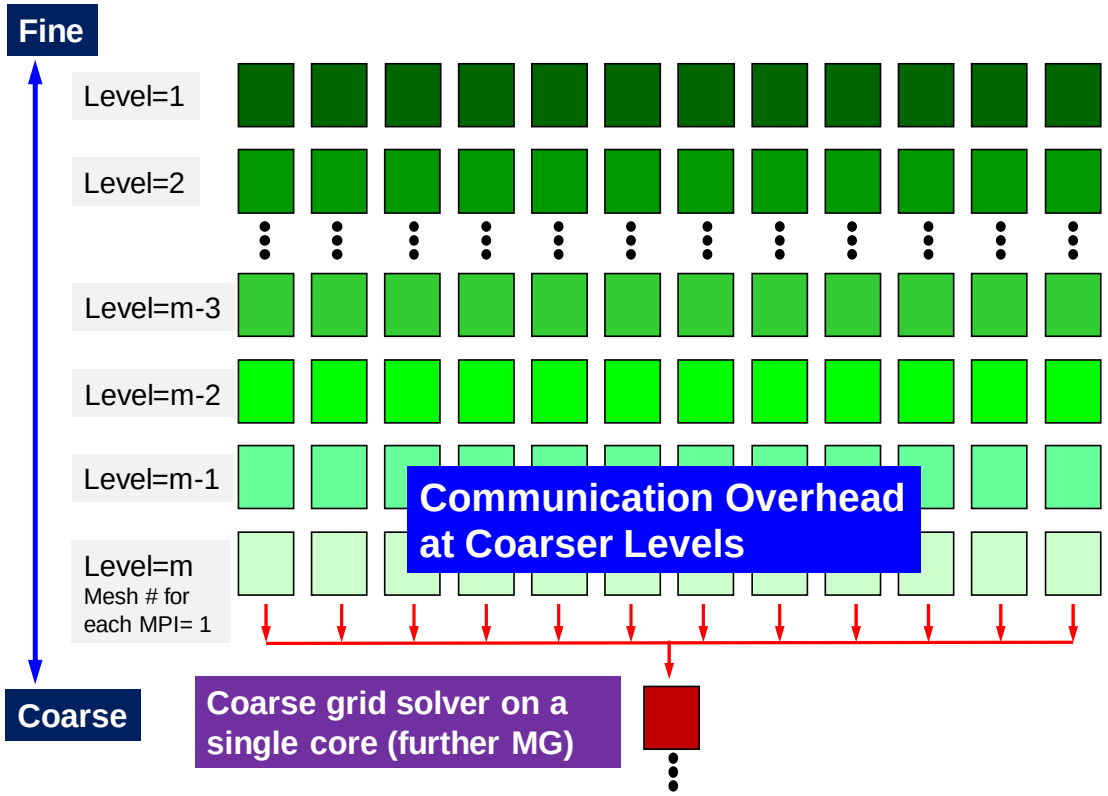
Overview: Highlights

- **AM-*h*CGA (Adaptive Multilevel-Hierarchical Coarse Grid Aggregation)** was proposed for large-scale multigrid methods on massively parallel systems

- AM-*h*CGA outperformed existing method (hCGA [KN 2014]) at $O(10^5)$ MPI processes

- This was done by using IHK/McKernel developed by RIKEN R-CCS [Gerofi, Ishikawa et al. IPDPS 2016]

Parallel Multigrid Method

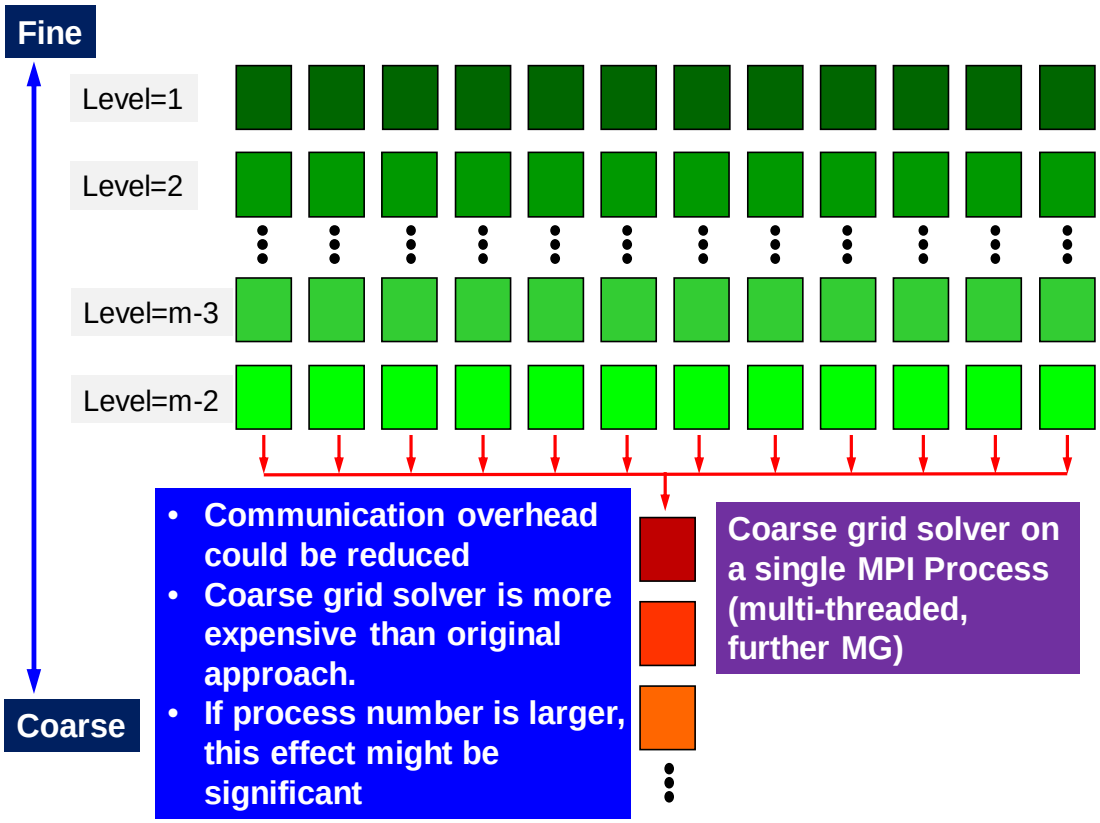


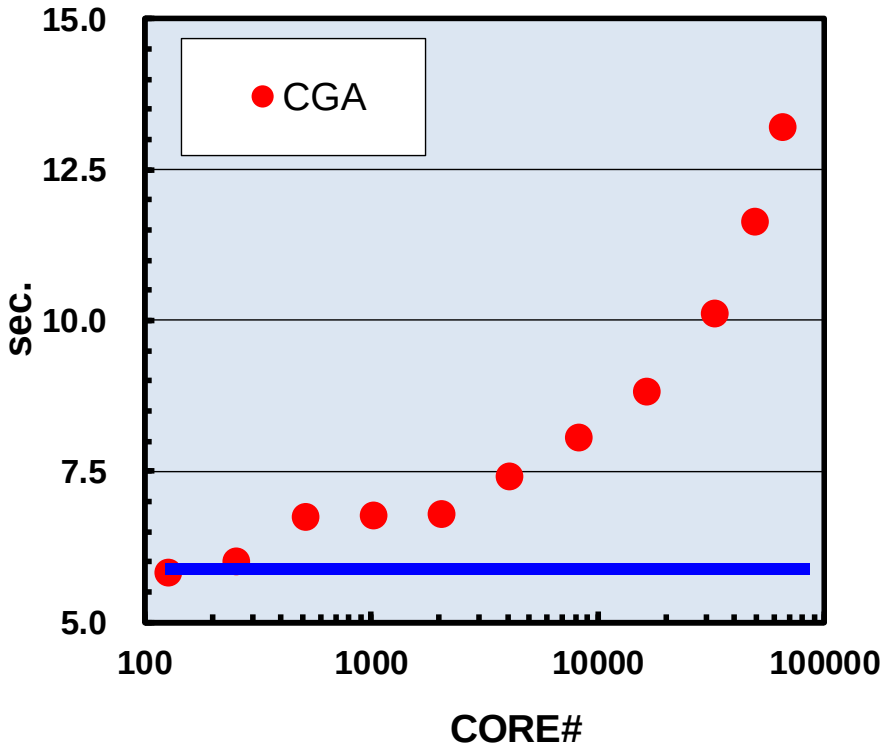
Communication Overhead at Coarse Levels

Coarse Grid Solver Serial Operations

Coarse Grid Aggregation (CGA) [KN 2012]

MGCG on Fujitsu FX10 up to 4,096 nodes, 17,179,869,184 DOF

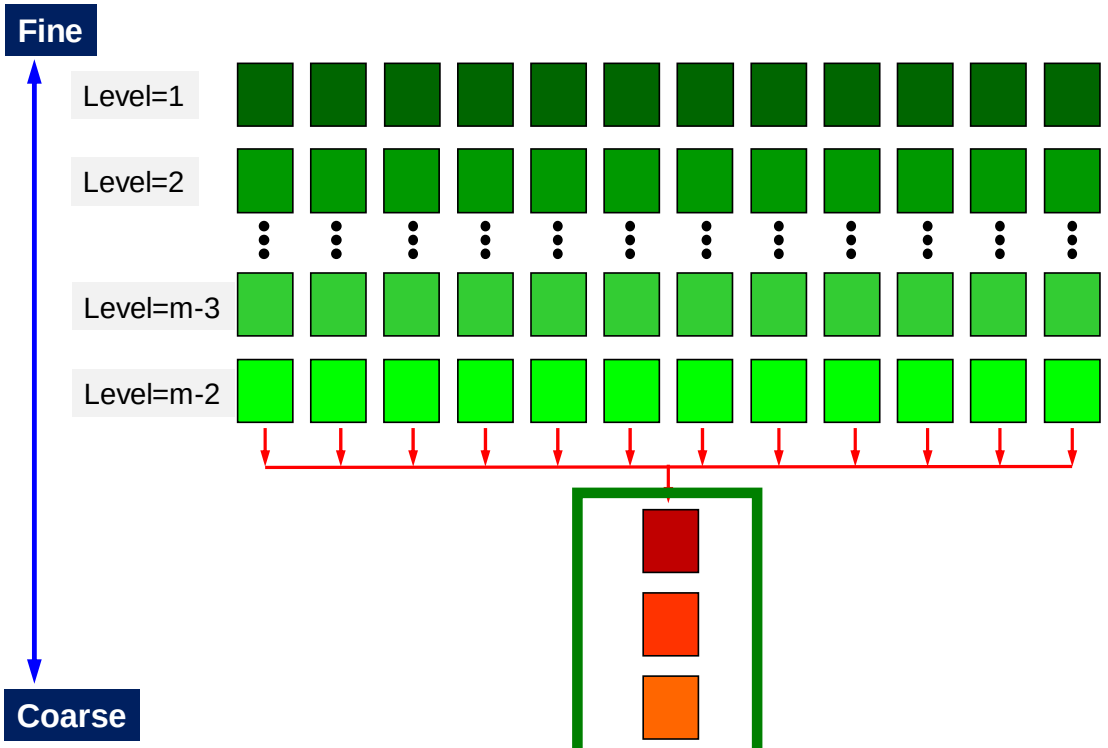




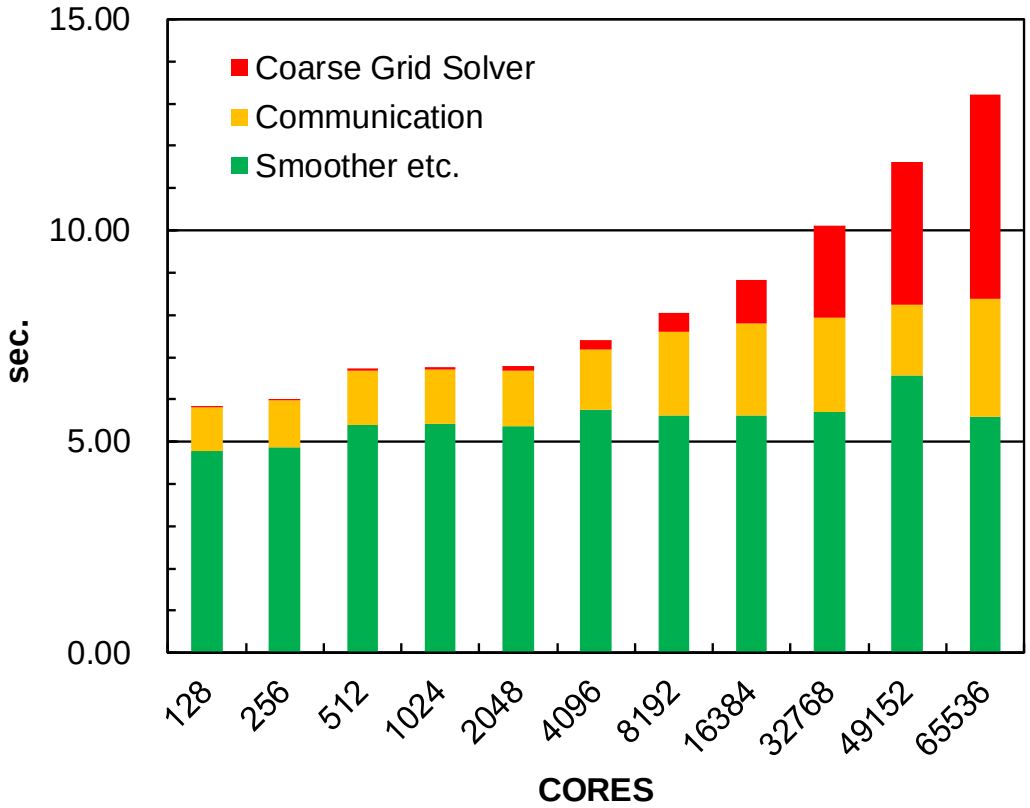
Weak Scaling: should be FLAT

Coarse Grid Aggregation (CGA) [KN 2012]

MGCG on Fujitsu FX10 up to 4,096 nodes, 17,179,869,184 DOF



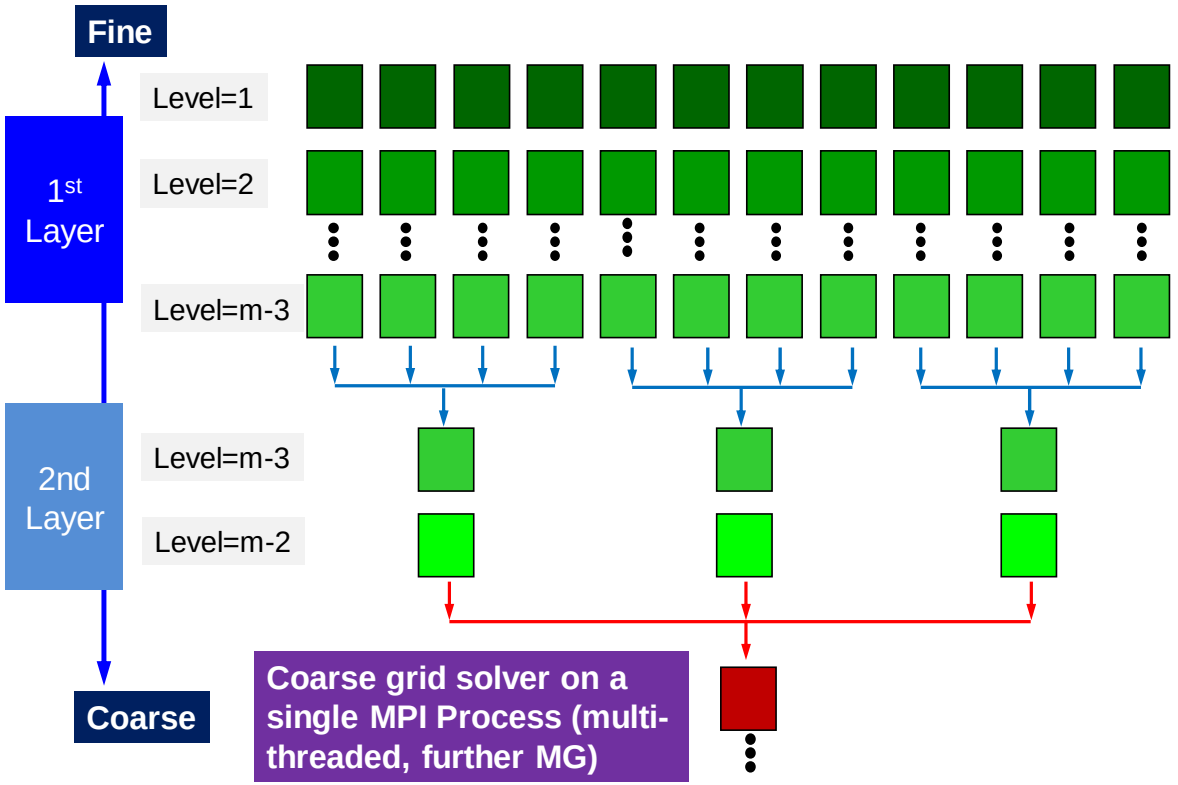
Cost of Coarse Grid Solver = Serial Operations

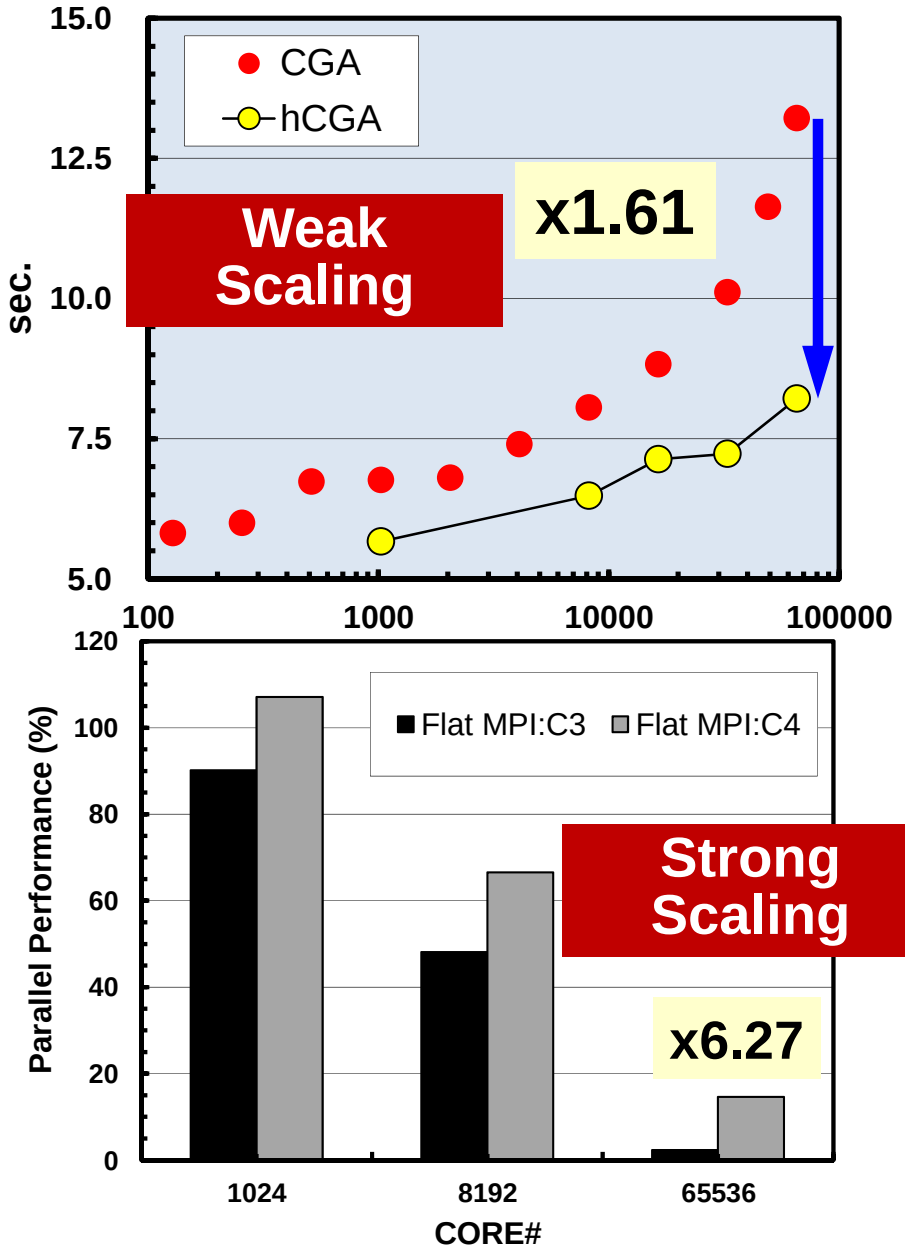


Cost of Coarse Grid Solver is significant if number of MPI processes is larger

Hierarchical CGA (*h*CGA) [KN 2014]

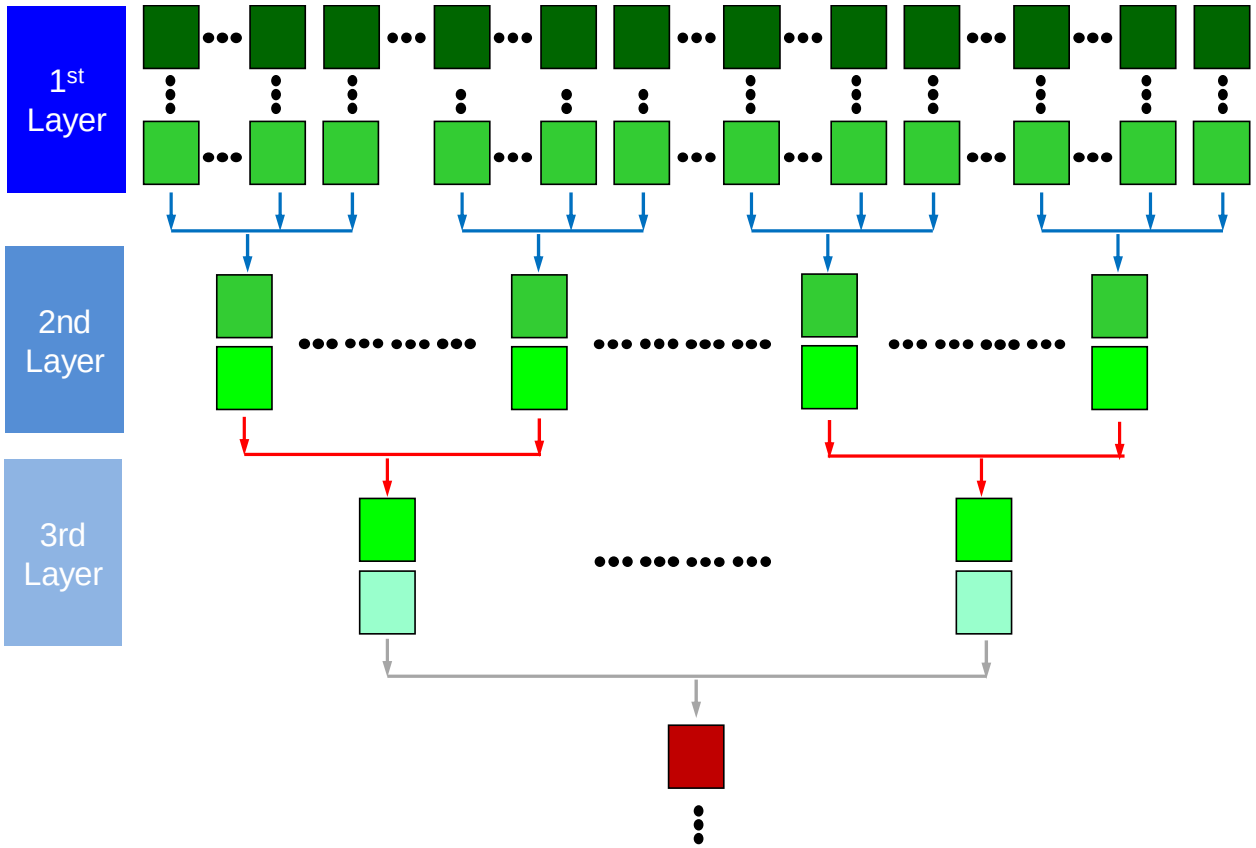
MGCG on Fujitsu FX10 up to 4,096 nodes, 17,179,869,184 DOF





Proposed Method: AM-*h*CGA

Adaptive Multilevel *h*CGA



- If the number of MPI processes is O(10^4), *h*CGA is effective
- If the number of MPI processes is O(10^6-10^7), number of processes at the 2nd level of *h*CGA could be O(10^4).
  - 2-Layers might not be enough for more processes
  - More levels are needed ?
- AM-*h*CGA
  - 3-Layers in this work

*h*CGA & AM-*h*CGA on OFP

- Evaluation of CGA, *h*CGA & AM-*h*CGA
  - Time for MGCG solver evaluated
- **Up to 2,048 Nodes of OFP, Weak Scaling**
  - **Flat MPI, 64 cores/node: MAX 131,072 Processes**
  - **Flat Mode, Only MC-DRAM used**
  - **5 runs for each case: the best one is adopted**
- IHK/McKernel
- Three Configuration of Problems

	Medium	Small	Tiny
Core	64x32x32 = 65,536	32x16x16 = 8,192	16x8x8 = 1,024
Node (64 cores)	4,194,304	524,288	65,536
MAX (2,048 nodes)	8,589,934,592	1,073,741,824	134,217,728

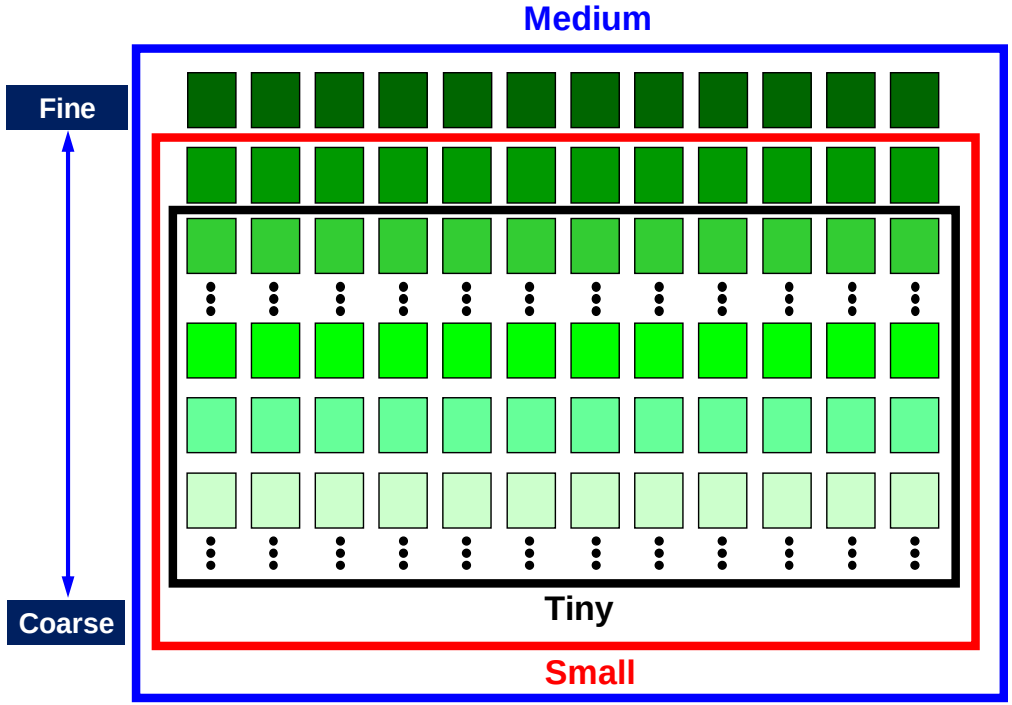
*h*CGA & AM-*h*CGA on OFP

- Evaluation of CGA, *h*CGA & AM-*h*CGA
  - Time for MGCG solver evaluated
- Up to 2,048 Nodes of OFP, Weak Scaling
  - **Flat MPI**, 64 cores/node: MAX 131,072 Process
  - Flat Mode, Only MC-DRAM used
  - 5 runs for each case: the best one is adopted
- IHK/McKernel
- Three Configuration of Problems

Flat MPI is adopted, mainly because we need huge number of MPI processes for evaluation of AM-*h*CGA

	Medium	Small	Tiny
Core	64x32x32 = 65,536	32x16x16 = 8,192	16x8x8 = 1,024
Node (64 cores)	4,194,304	524,288	65,536
MAX (2,048 nodes)	8,589,934,592	1,073,741,824	134,217,728

*h*CGA & AM-*h*CGA on OFP



Medium⇒Small⇒Tiny Effects of the coarse/coarser grid solver are more siginificant.

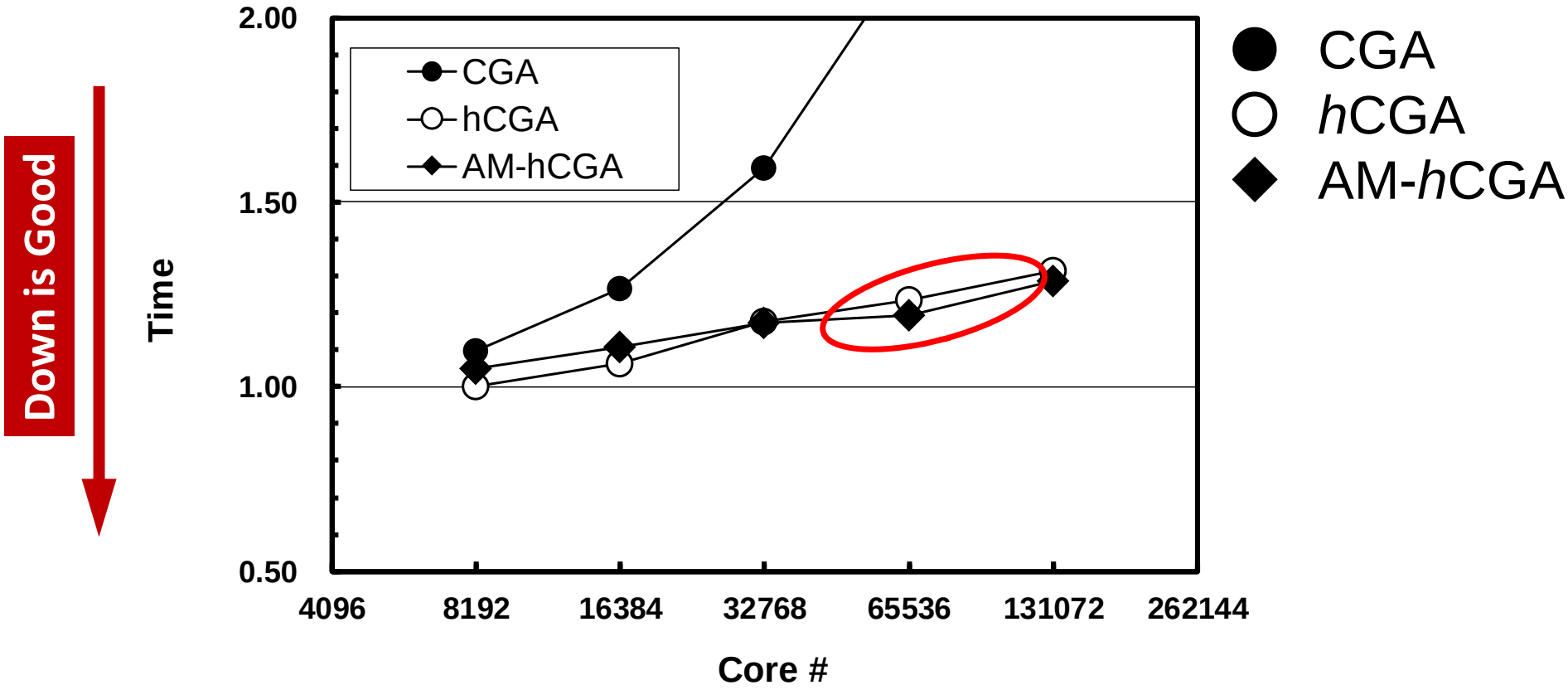
AM-*h*CGA is expected to work better in “Tiny” cases

	Medium	Small	Tiny
Core	64x32x32 = 65,536	32x16x16 = 8,192	16x8x8 = 1,024
Node (64 cores)	4,194,304	524,288	65,536
MAX (2,048 nodes)	8,589,934,592	1,073,741,824	134,217,728

Preliminary Results of AM-*h*CGA, Weak Scaling Max. 8,588,824,592 DOF (Medium)

Time for CGA with 128 nodes (8,192 cores)= 1.00

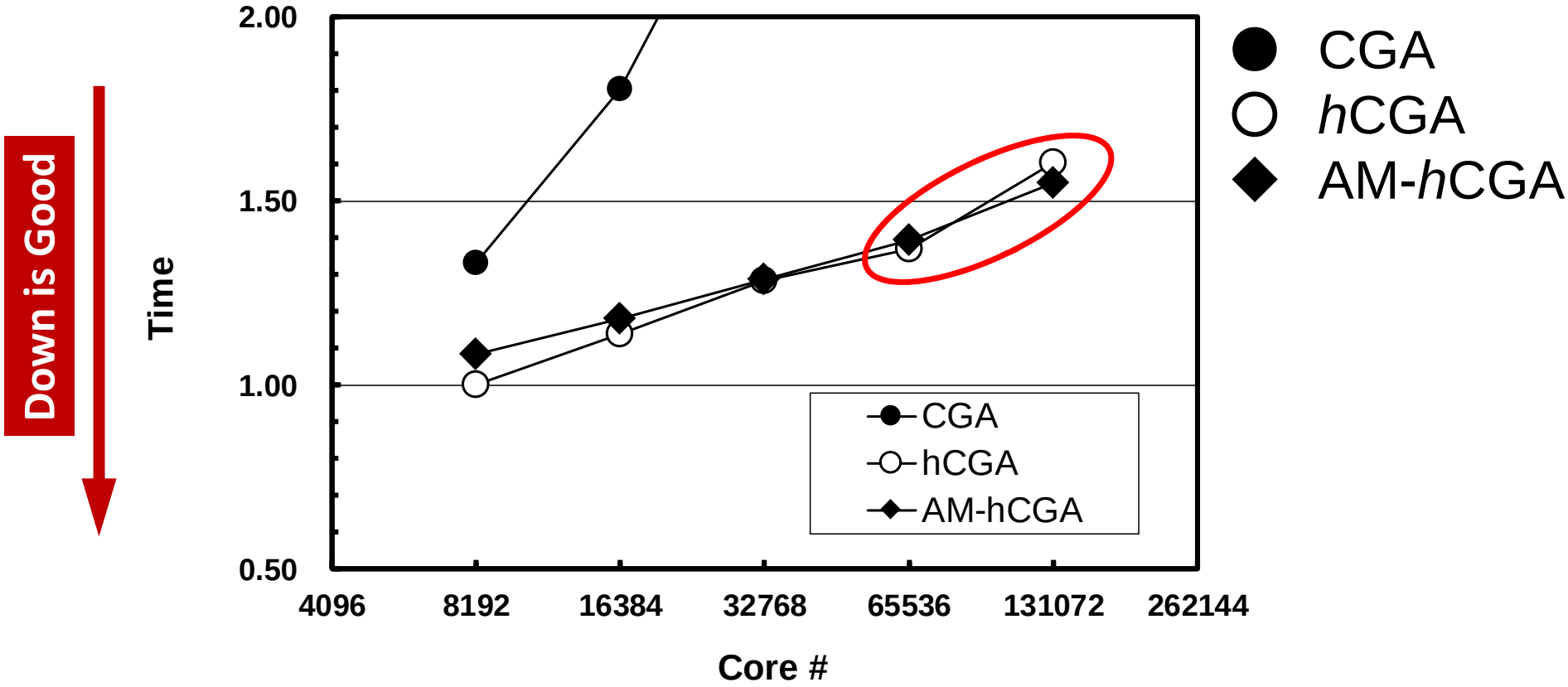
AM-hCGA is slightly better at O(10^5) cores



Preliminary Results of AM-*h*CGA, Weak Scaling Max. 1,073,741,824 DOF (Small)

Time for CGA with 128 nodes (8,192 cores)= 1.00

AM-hCGA is slightly better at O(10^5) cores, more obvious than "Med"

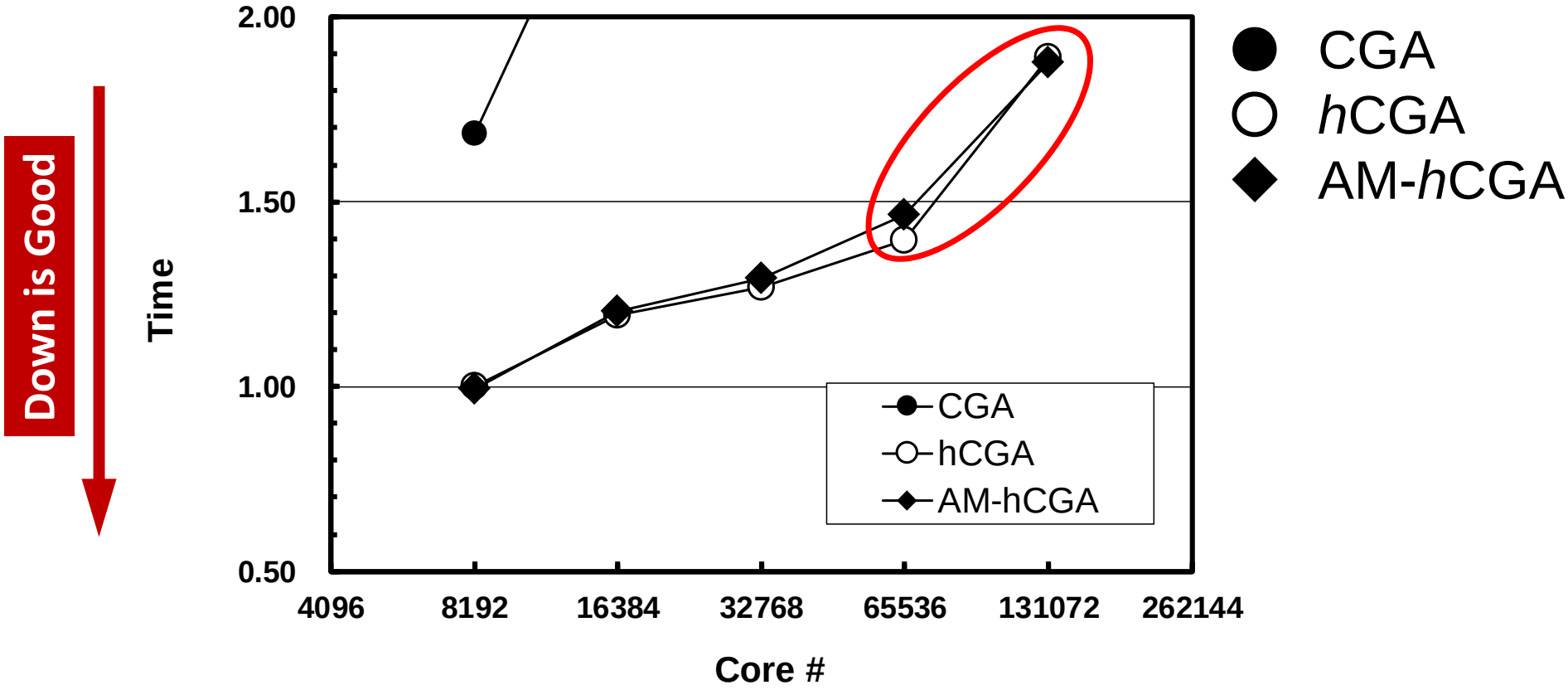


Preliminary Results of AM-*h*CGA, Weak Scaling

Max. 1,073,741,824 DOF (Tiny)

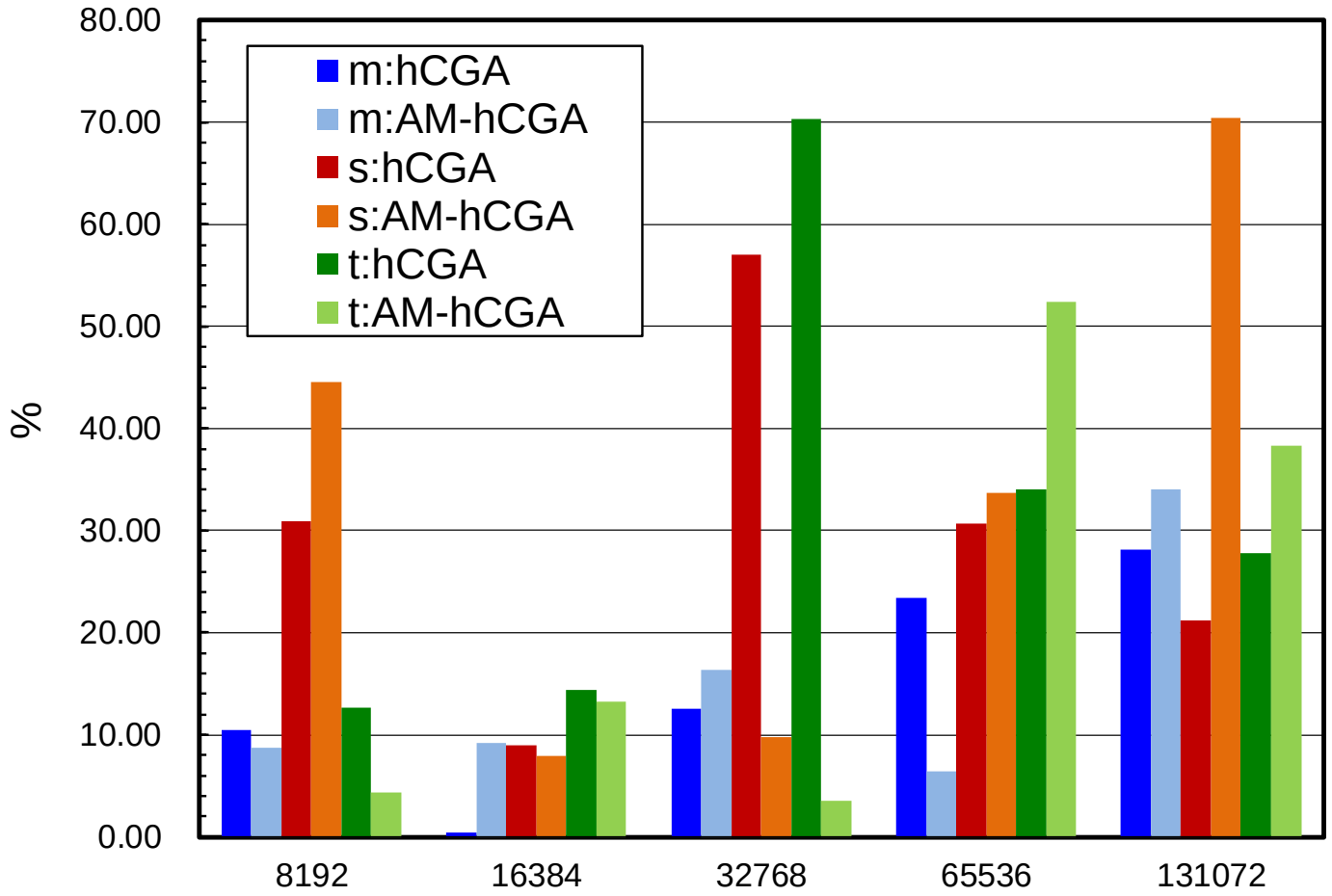
Time for CGA with 128 nodes (8,192 cores)= 1.00

Larger effects of AM-hCGA are expected than "Small" cases, but …



Fluctuation of 5 Measurements

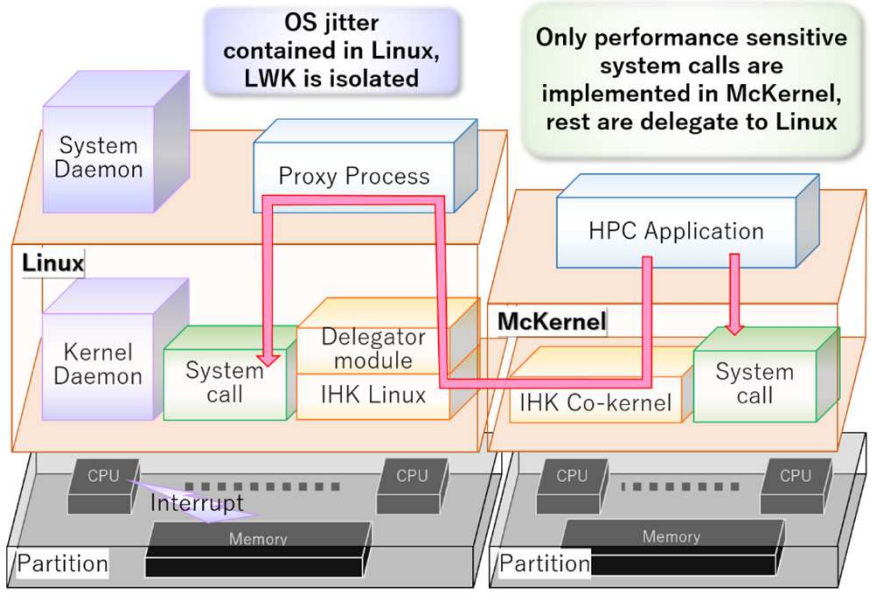
Effects of OS Jitter etc. are significant for larger number of nodes
It is unclear whether AM-*h*CGA is really faster than *h*CGA, or not.



$$100 \times \left(\frac{T_{max} - T_{min}}{T_{min}} \right)$$

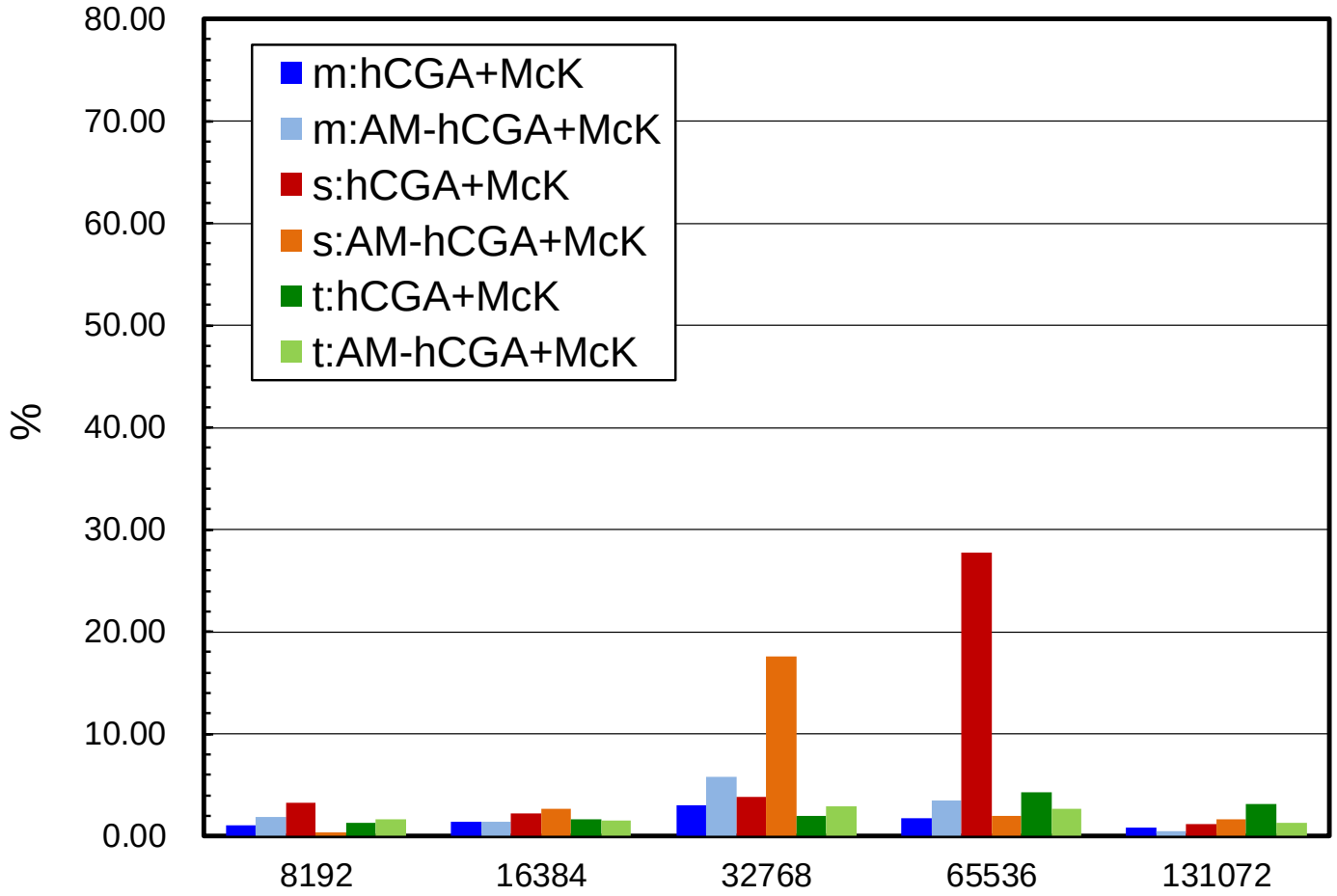
IHK/McKernel [Gerofi et al. IPDPS 2016]

- Lightweight Multi-Kernel OS for HPC by RIKEN R-CCS
- McKernel implements only a small set of performance sensitive system calls and the rest of the OS services are delegated to Linux
  - Linux + McKernel
- Same binary on pure Linux can be used
- Lower Noise/Communication Overhead than the Pure Linux Environment
- to be installed on the Fugaku (Post K)
  - Already on K and OFP



IHK/McKernel applied !!

There are still certain fluctuations, but much better than before



$$100 \times \left(\frac{T_{max} - T_{min}}{T_{min}} \right)$$

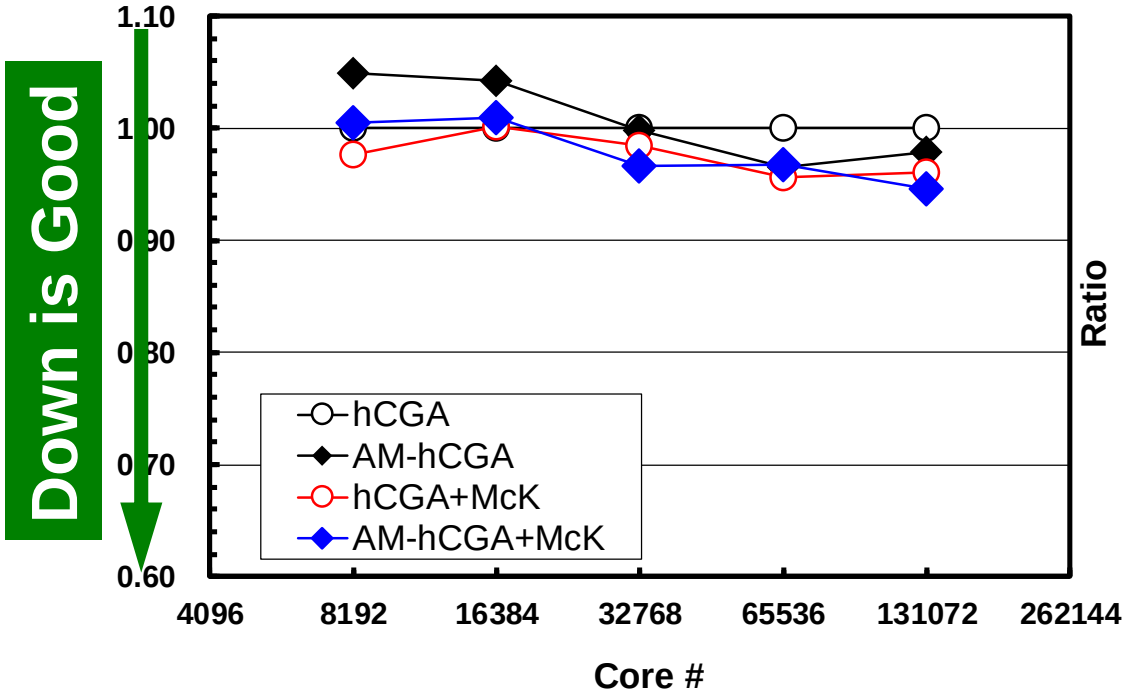
○hCGA+McK~◆AM-hCGA+McK (Stable Case)

◆ is slightly better at O(10^5) cores

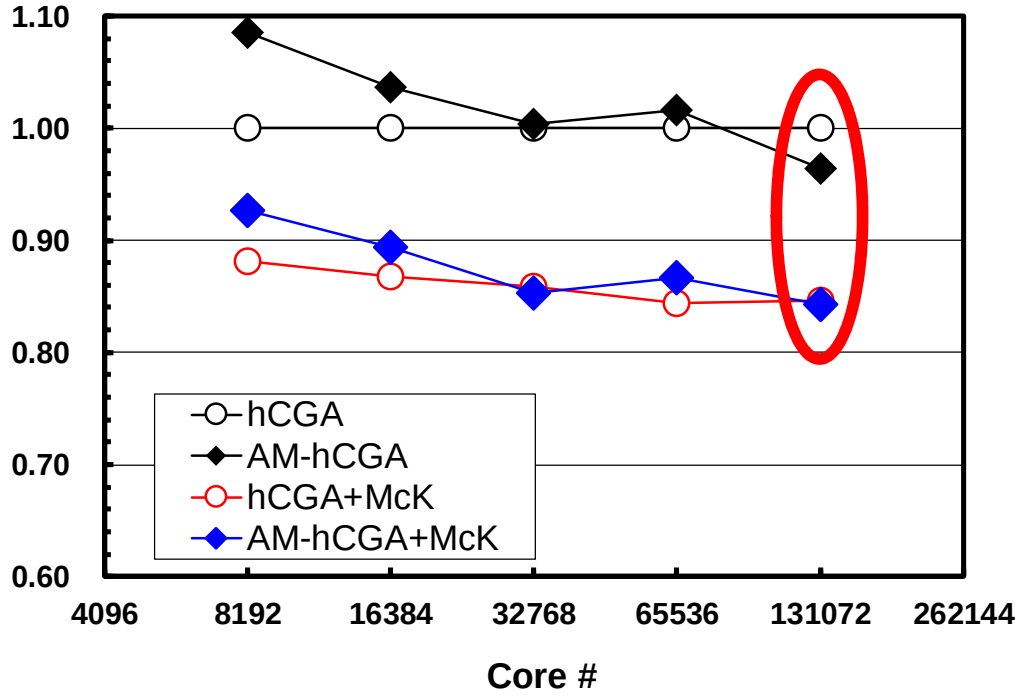
15% Improvement of Performance in "Small" case by IHK/McKernel

Computation time is normalized by that of *h*CGA with 8,192 cores

Medium: 64x32x32 DOF/core



Small: 32x16x16 DOF/core



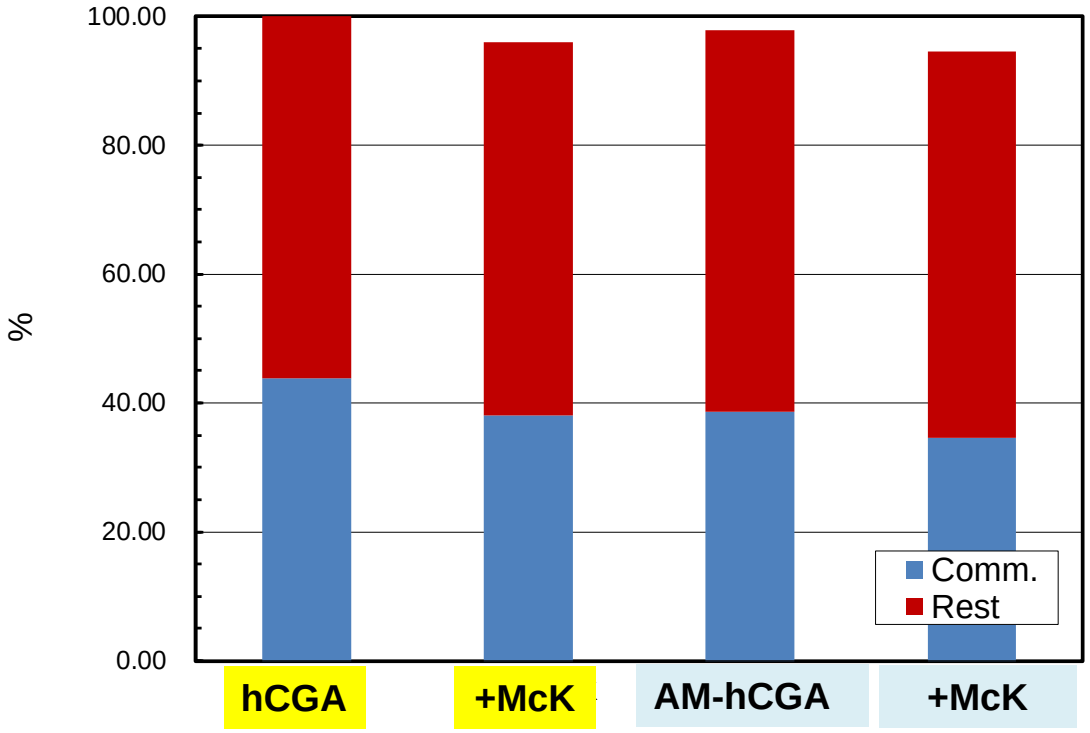
○ hCGA (=1.00) ◆ AM-hCGA ○ hCGA+McK ◆ AM-hCGA+McK

AM-*h*CGA: Effects of IHK/McKernel

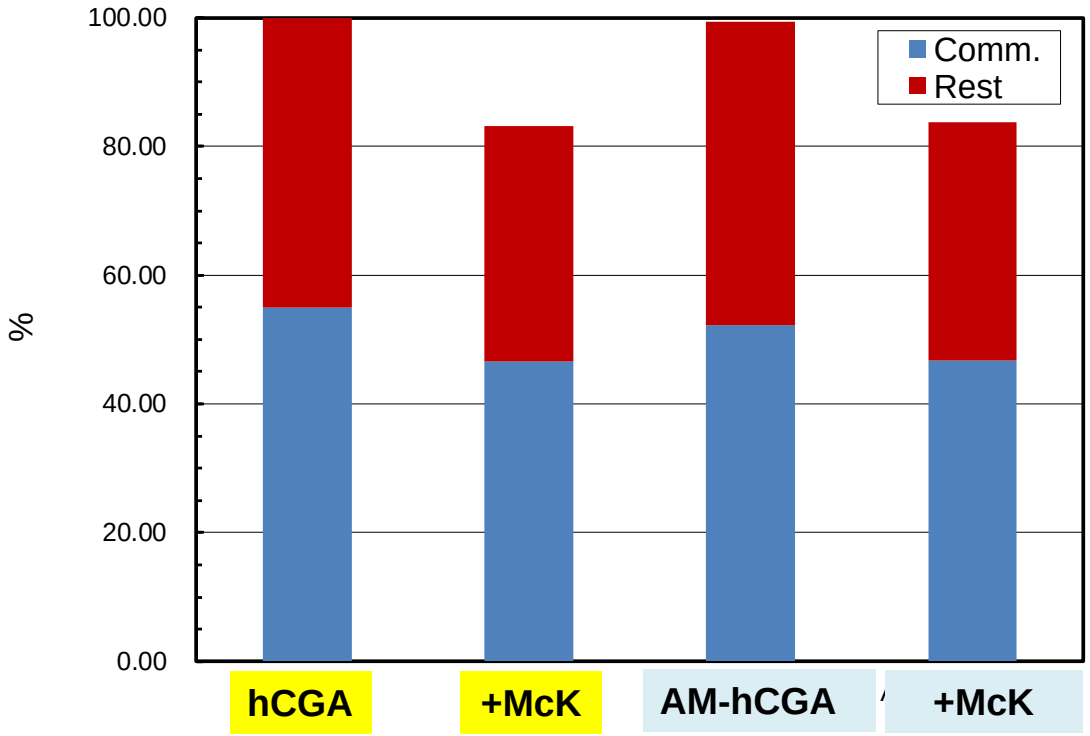
Computation Time for hCGA= 100%, Communication, Others

Significant reduction of comm. time by IHK/McKernel in "Small" case

Medium: 64x32x32 DOF/core

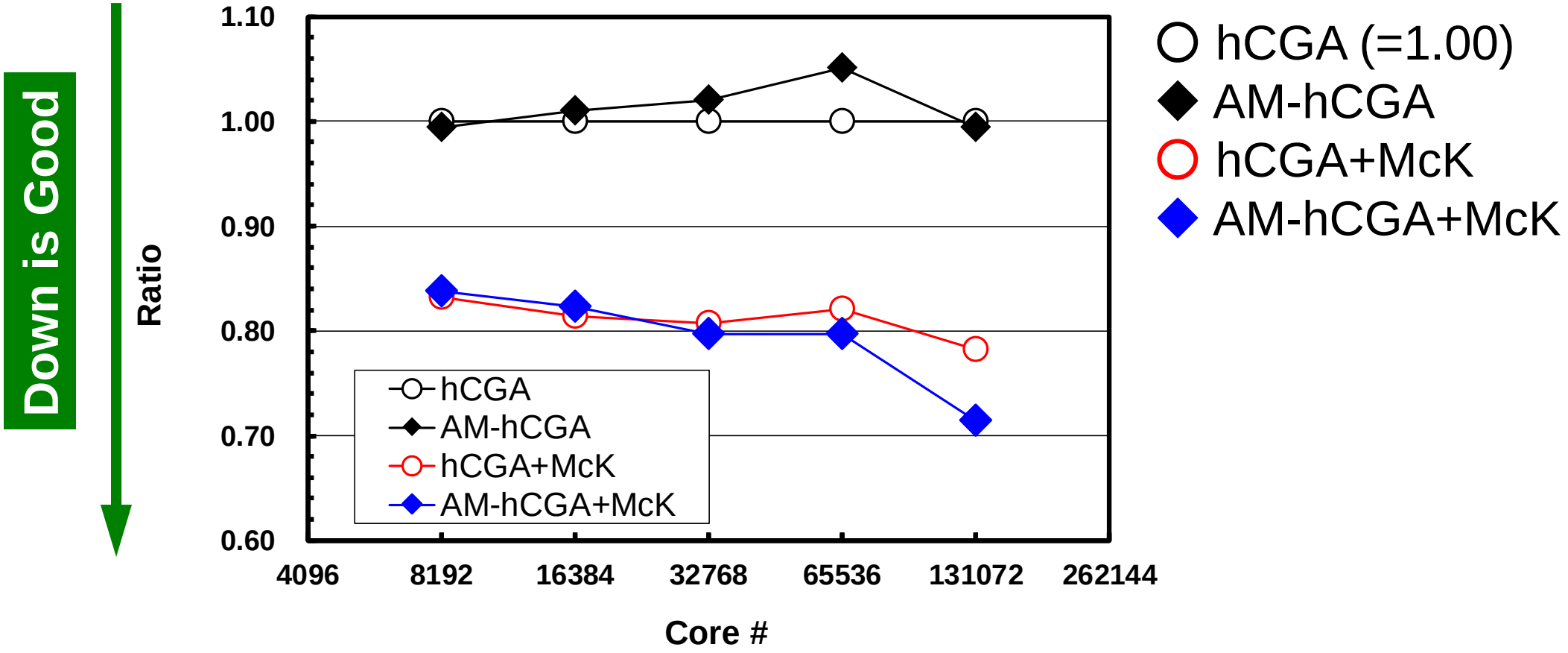


Small: 32x16x16 DOF/core



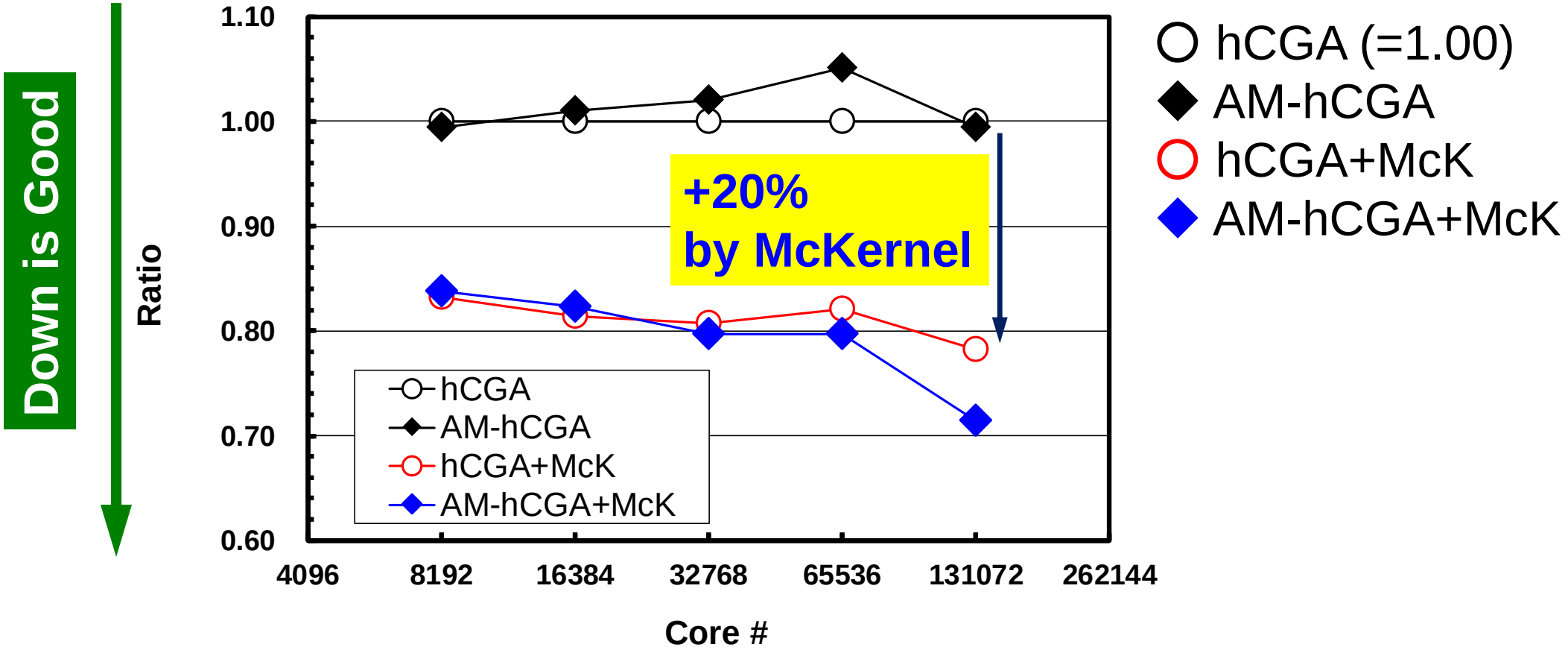
“Tiny” Cases: More Siginicant Improvement by IHK/McKernel and AM-*h*CGA

Computation time is normalized by that of *h*CGA



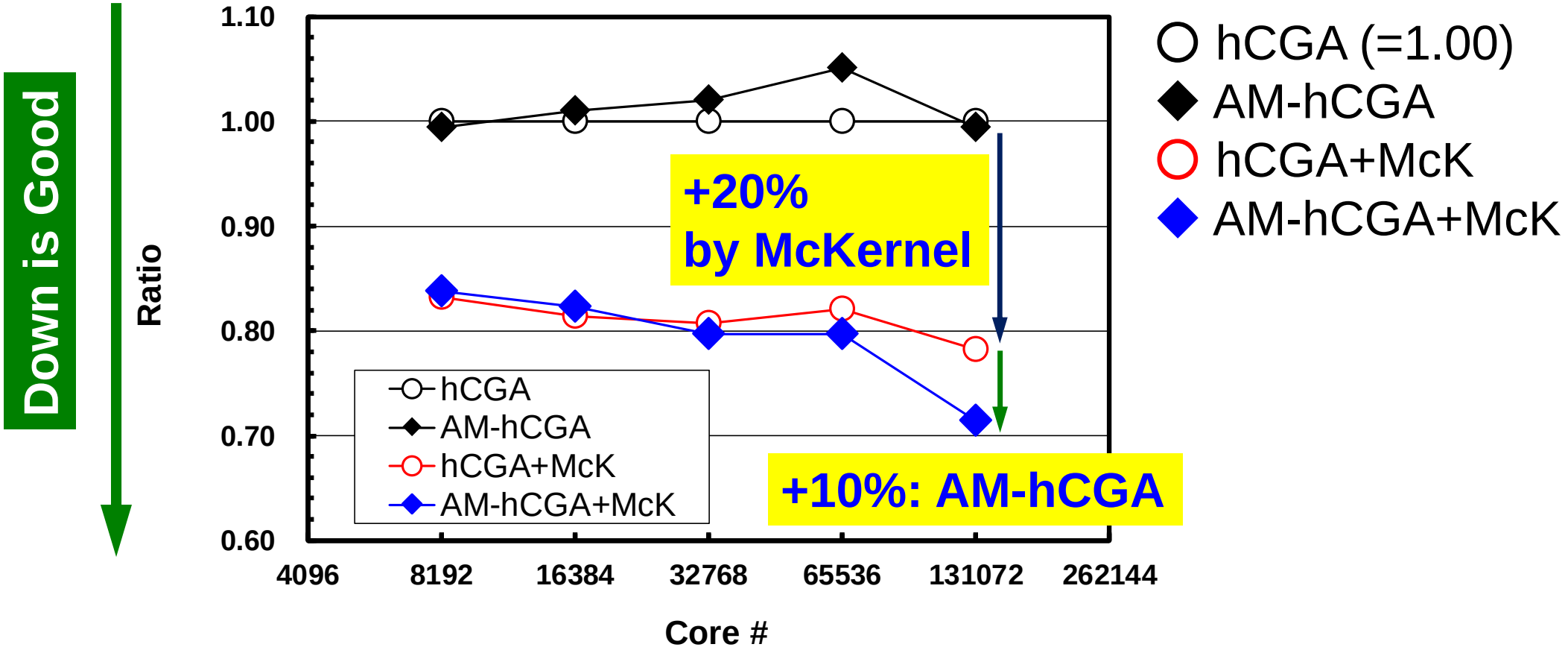
“Tiny” Cases: More Siginicant Improvement by IHK/McKernel and AM-*h*CGA

+20% by IHK/McKernel



“Tiny” Cases: More Siginicant Improvement by IHK/McKernel and AM-*h*CGA

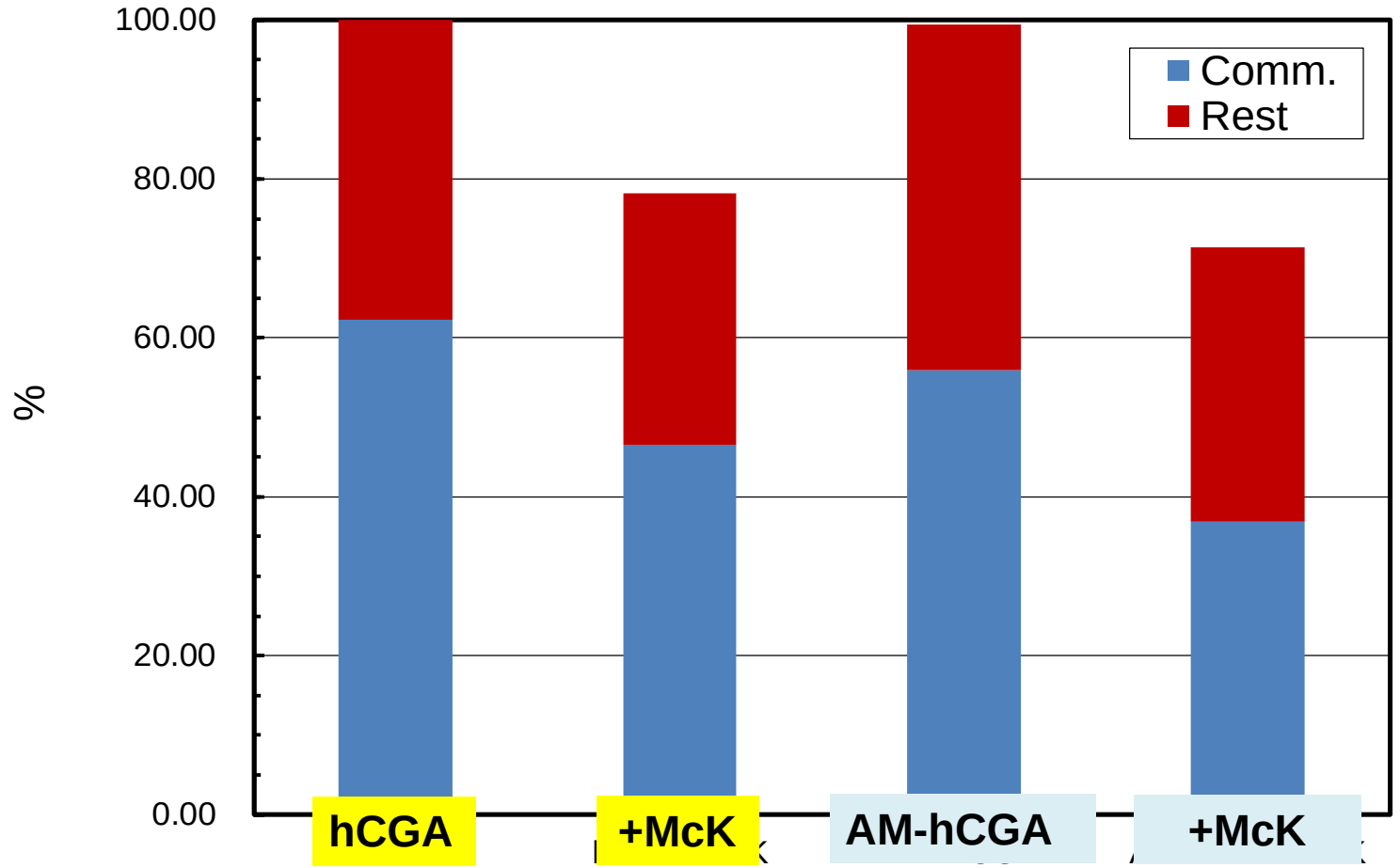
+20% by IHK/McKernel, +10% by AM-*h*CGA



AM-*h*CGA: Effects of IHK/McKernel

Computation Time for hCGA= 100%, Communication, Others

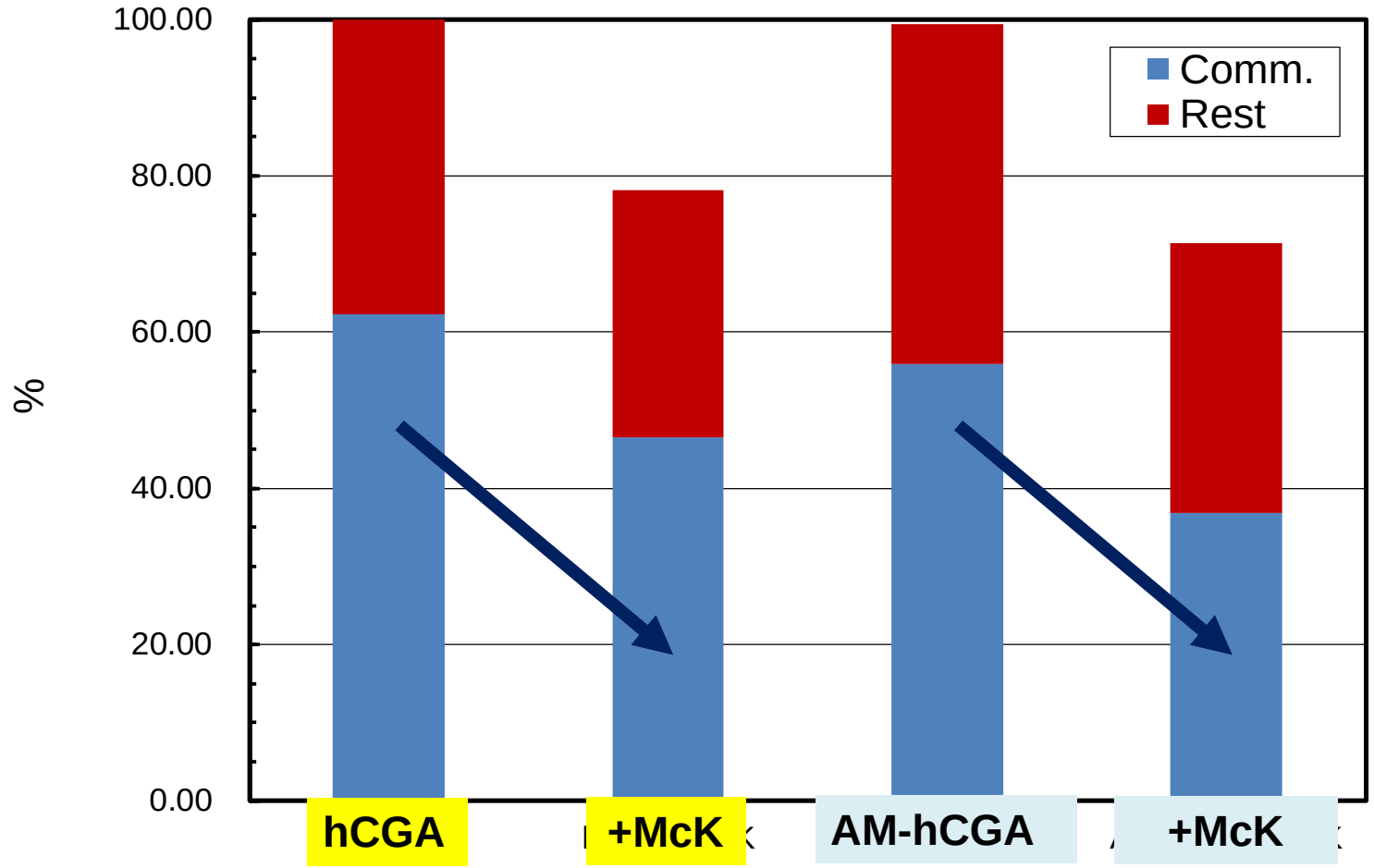
More significant reduction of comm. time by IHK/McKernel in "Small"



AM-*h*CGA: Effects of IHK/McKernel

Computation Time for hCGA= 100%, Communication, Others

More significant reduction of comm. time by IHK/McKernel in "Small"



Summary

- Adaptive Multilevel *h*CGA (AM-*h*CGA) for Optimization of Parallel GMG
- AM-*h*CGA may attain better performance if the number of MPI processes are O(10^5), and problem size/core is small
  - 2 levels for O(10^4), 3 levels for O(10^5), 4 levels for (10^6) ?
- Because fluctuation of performance on OFP with many nodes is significant, IHK/McKernel is very effective
  - Significance of AM-*h*CGA was not proved without McKernel
  - McKernel is essential for using OFP with O(10^3) nodes !!
- Future Works
  - Pipelined Algorithms
  - SELL-C-σ
  - Lower/Mixed Precision
  - Larger Problems using More Cores