Parallel Multigrid Method on Multicore/Manycore Clusters

Kengo Nakajima
Information Technology Center, The University of Tokyo
RIKEN R-CCS

IXPUG HPC Asia 2020
January 17, 2020, Fukuoka, Japan

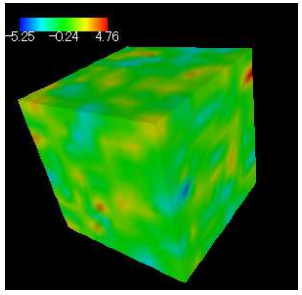
Acknowledgements

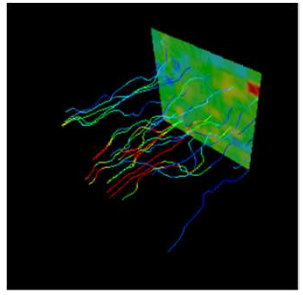
- JST/CREST
- DFG/SPPEXA

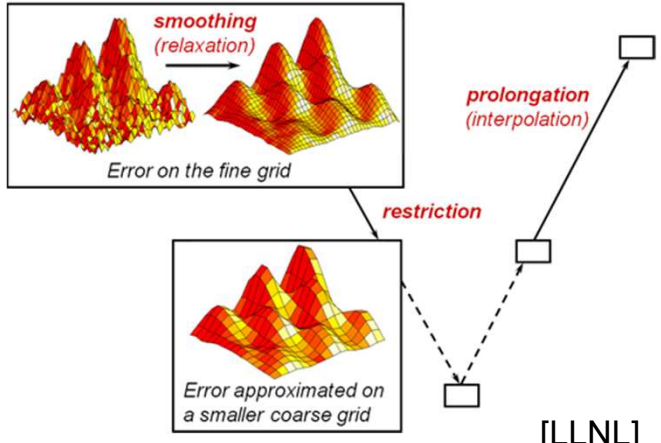
- **JHPCN (jh180022-NAHI, jh180041-NAHI)**
  - **Innovative Multigrid Methods**
- JCAHPC
  - Large-Scale HPC Challenge on Oakforest-PACS
- **Balazs Gerofi (RIKEN R-CCS)**
- **Yutaka Ishikawa (RIKEN R-CCS)**
- **Masashi Horikoshi (Intel)**
- **Yoshio Sakaguchi (Fujitsu)**

Target Application

- 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 Iterative Solver preconditioned by Multigrid (MGCG), Geometric MG, IC(0) Smoother, V-cycle**
  - Sliced ELL for Storage of Sparse Matrices
- **Multigrid**
  - **Scalable O(N) algorithm, but many problems towards Exascale Computing**







[LLNL]

Previous Work focusing on Oakforest-PACS (OFP, Intel Xeon Phi (KNL) Cluster) [KN ScalA19]

Parallel Multigrid Methods on Manycore Clusters with IHK/McKernel

- Kengo Nakajima, Balazs Gerofi, Yutaka Ishikawa, Masashi Horikoshi
- ScalA19: 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems in conjunction with SC19, November 18, 2019, Denver, CO

Mostly, we used the code developed in this previous work [KN ScalA19]

Overview: Highlights

- **AM-*h*CGA (Adaptive Multilevel-Hierarchical Coarse Grid Aggregation)** for large-scale multigrid methods on massively parallel systems [KN ScalA19]
- Performance Evaluations of CGA/*h*CGA/AM-*h*CGA on the Following Platforms using up to 2,048 Sockets
  - Oakforest-PACS (OFP)
    - Intel Xeon Phi (Knights Landing, KNL), Fujitsu
    - IHK/McKernel
    - 8,208 nodes, 25+PF, 15th in TOP 500 (Nov.2019)
    - Operated by JCAHPC (U.Tsukuba & U.Tokyo)
  - Oakbridge-CX (OBCX)
    - Intel Platinum 8280 (Cascade Lake, CLX), Fujitsu
    - 1,368 nodes (2,736 sockets), 6.61 PF, 50th in TOP 500 (Nov.2019)
  - **Code(s): Optimized for OFP in [KN ScalA19] (not fully)**
  - **Significant Performance on OBCX for Strong Scaling**

JCAHPC

東京大学 THE UNIVERSITY OF TOKYO

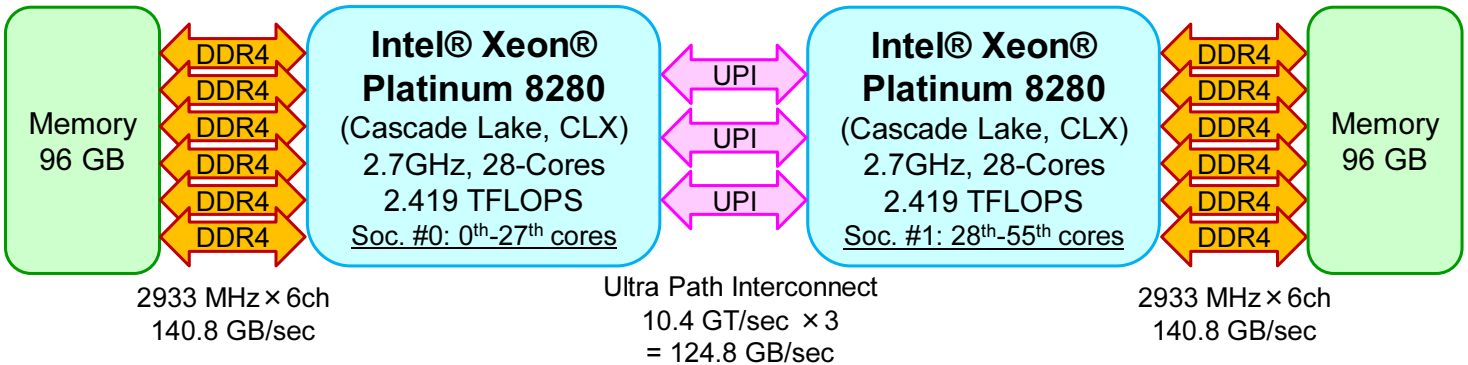
筑波大学 University of Tsukuba

Overview of Each Socket

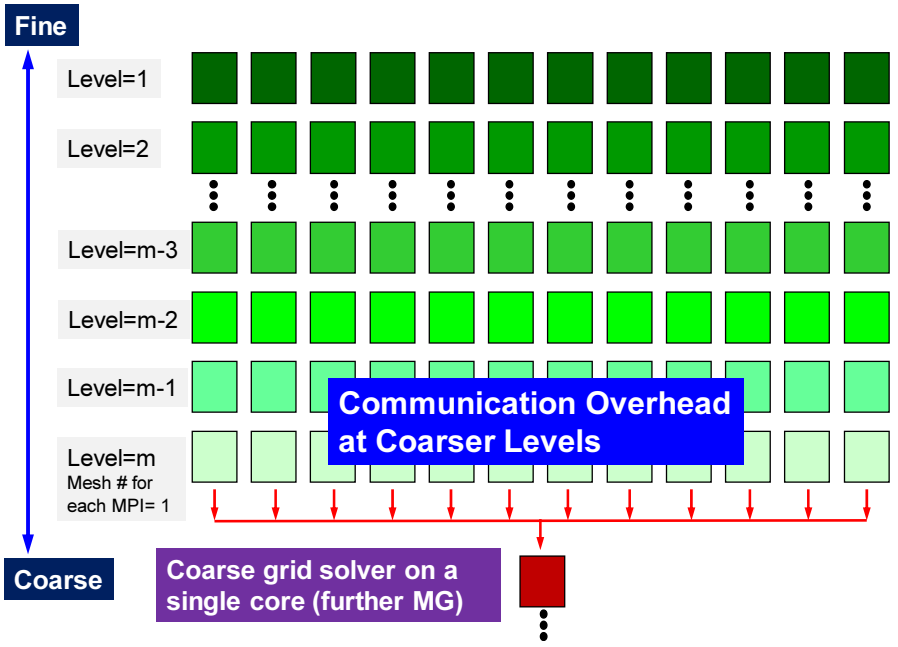
System	Oakforest-PACS (OFP)	Oakbridge-CX (OBCX)
Name in this Paper	OFP	OBCX
Architecture	Intel Xeon Phi 7250 (Knights Landing, KNL)	Intel Xeon Platinum 8280 (Cascade Lake, CLX)
Frequency (GHz)	1.40	2.70
Core #/CPU (socket)	**68**	**28**
CPU (socket) # per node	**1**	**2**
Peak Performance (GFLOPS) per socket	**3,046.4**	**2,419.2**
Memory Size (GB） per socket	MCDRAM: 16 DDR4: 96	96
Memory Bandwidth/Socket (GB/sec, STREAM Triad)	**MCDRAM: 490 DDR4: 84.5**	**101.0**
Peak Performance per Core (GFLOPS)	44.8	86.4
Memory Bandwidth per Core (GB/sec., STREAM Triad)	MCDRAM: 7.21 DDR4: 1.24	3.61

Overview of Each Socket

System	Oakforest-PACS (OFP)	Oakbridge-CX (OBCX)
Core #/CPU (socket)	68	**28**
CPU (socket) # per node	1	**2**
Peak Performance (GFLOPS) per socket	3,046.4	**2,419.2**
Memory Size (GB） per socket	MCDRAM: 16 DDR4: 96	96
Memory Bandwidth/Socket (GB/sec, STREAM Triad)	MCDRAM: 490 DDR4: 84.5	**101.0**



Parallel Multigrid Method

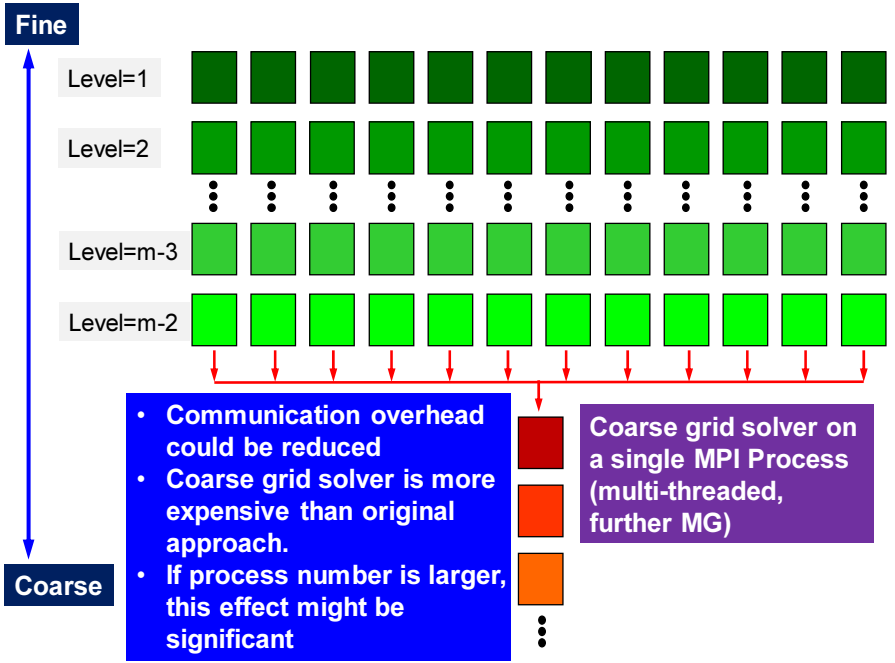


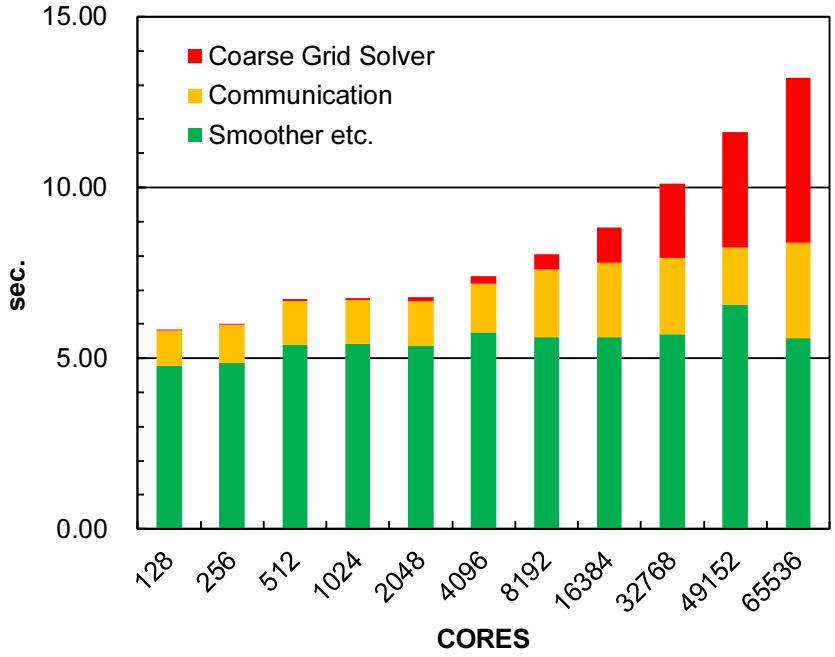
Communication Overhead at Coarse Levels

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

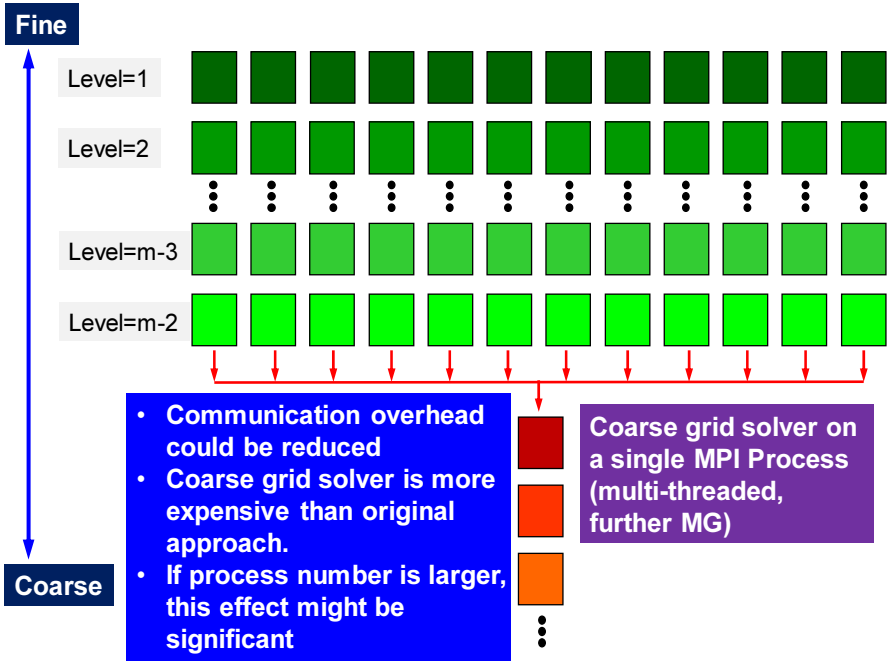


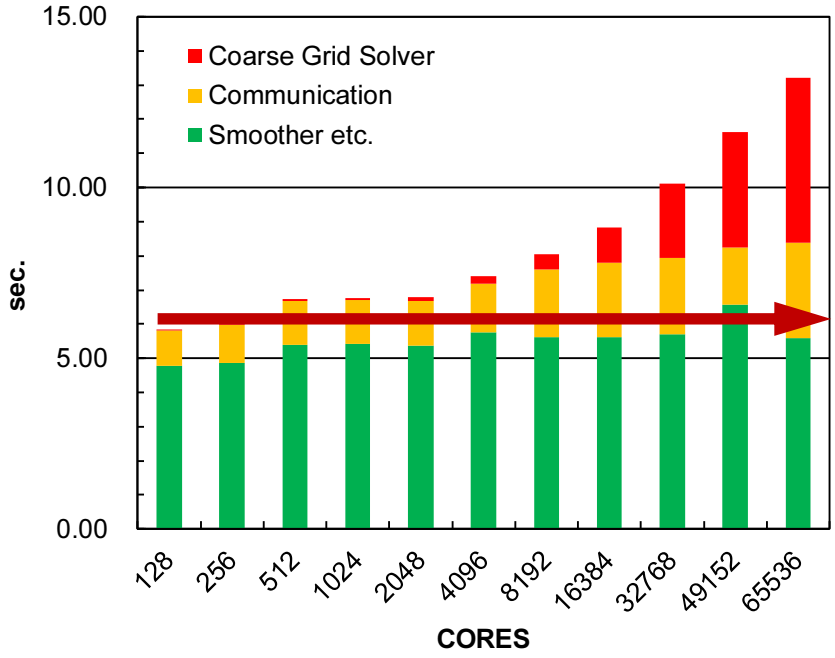


Weak Scaling: should be FLAT

Coarse Grid Aggregation (CGA) [KN 2012]

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

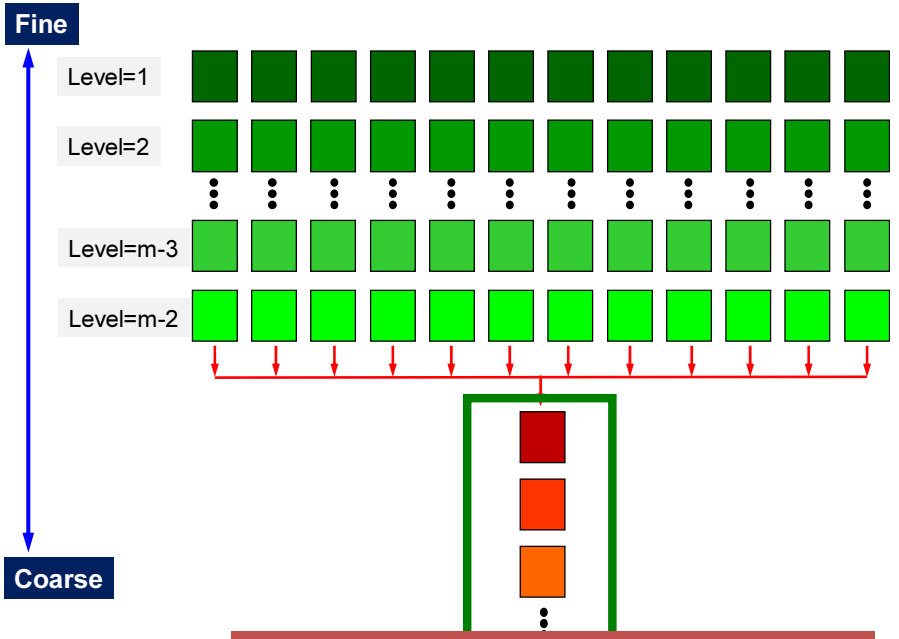




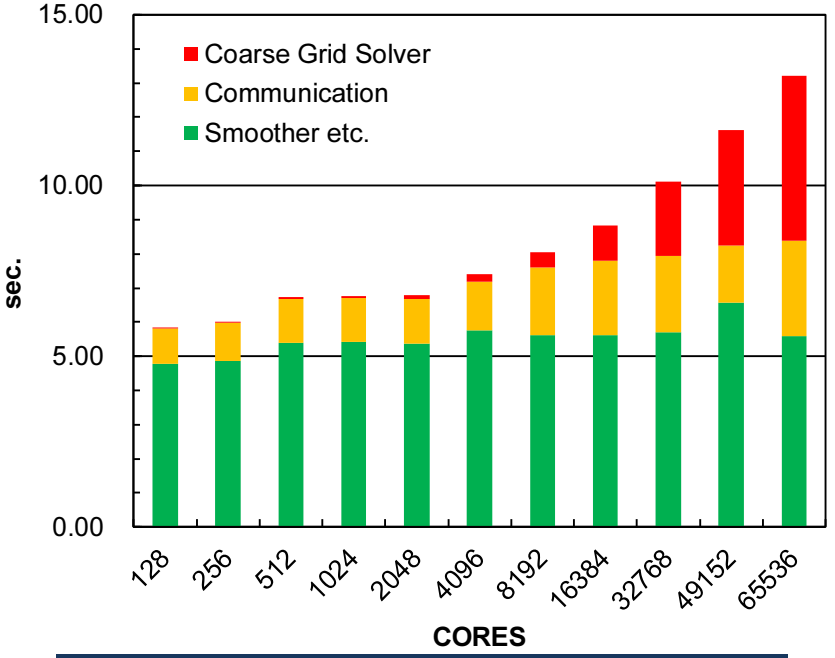
Weak Scaling: should be FLAT

Coarse Grid Aggregation (CGA) [KN 2012]

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



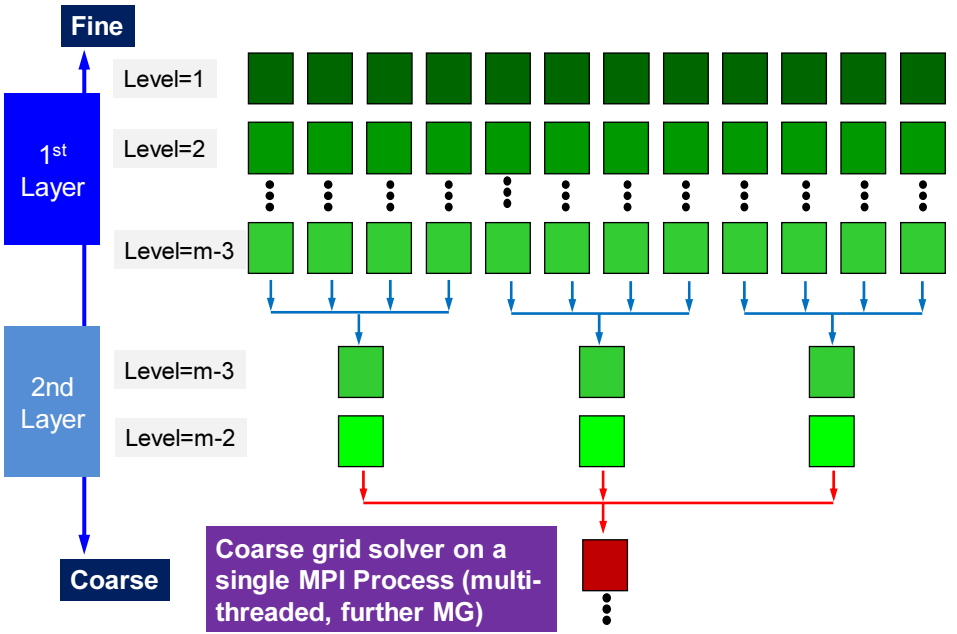
Cost of Coarse Grid Solver = Serial Operations

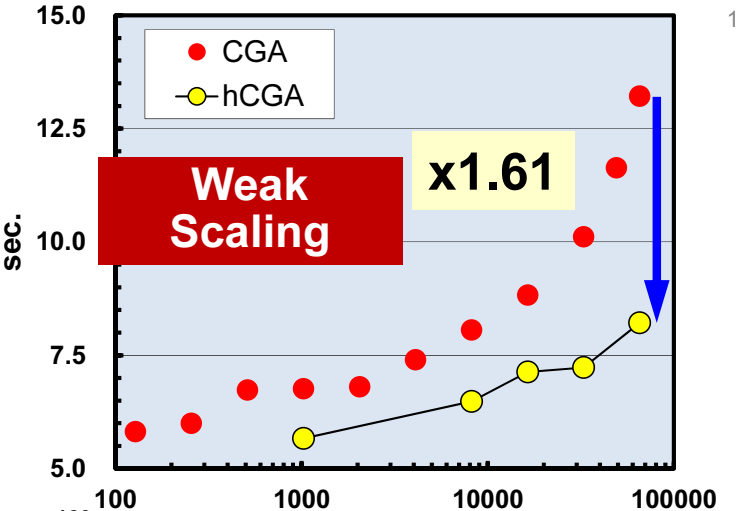


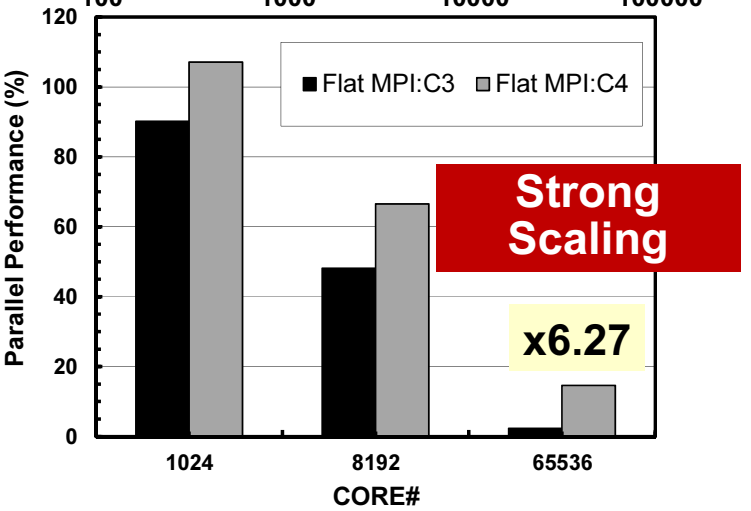
Cost of Coarse Grid Solver is significant if number of MPI processes is larger

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

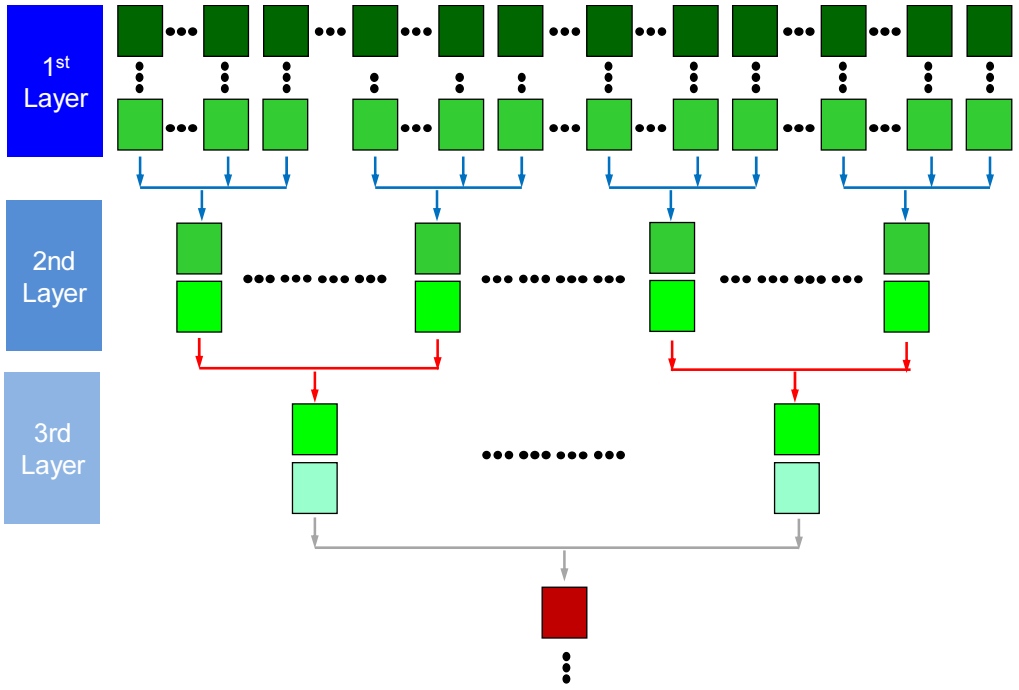
MGCG on Fujitsu FX10 up to 4,096 nodes, 17,179,869,184 DOF







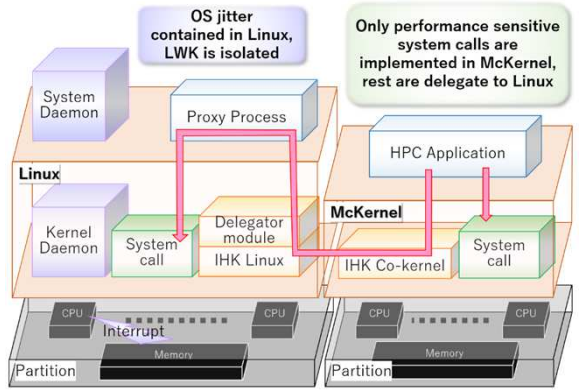
AM-*h*CGA: Adaptive Multilevel *h*CGA [KN ScalA19]



- If the number of MPI processes is $O(10^4)$, *h*CGA is effective
- If the number of MPI processes is $O(10^6\text{-}10^7)$, 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 [KN ScalA19]

- 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 (Last Talk)**
  - **Light Weight OS Kernel**
    - **Linux + IHK/McKernel**
    - **[Gerofi et al. IPDPS 2016]**
  - **Lower Noise**
  - **Lower Communication Overhead**
- 3 Configurations of Problems

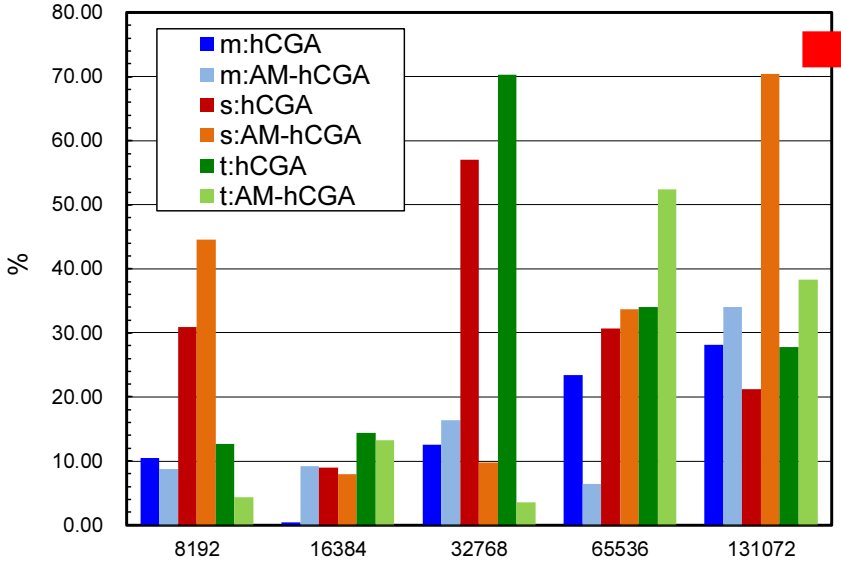


	Medium	Small	Tiny
Core	64x32x32	32x16x16	16x8x8
Node (64 cores)	4,194,304	524,288	65,536
MAX (2,048 nodes)	8,589,934,592	1,073,741,824	134,217,728

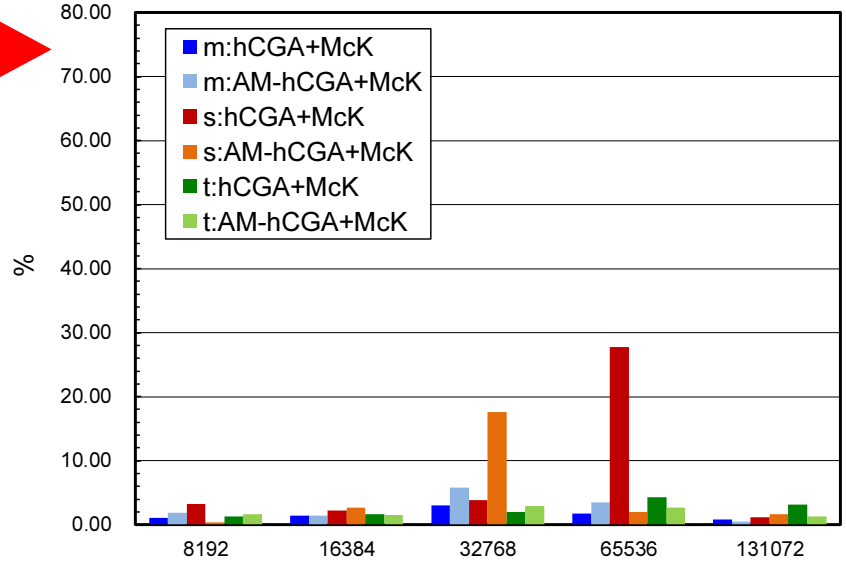
Fluctuation of 5 Measurements on OFP [KN ScalA19]

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

Without IHK/McKernel
Only Linux

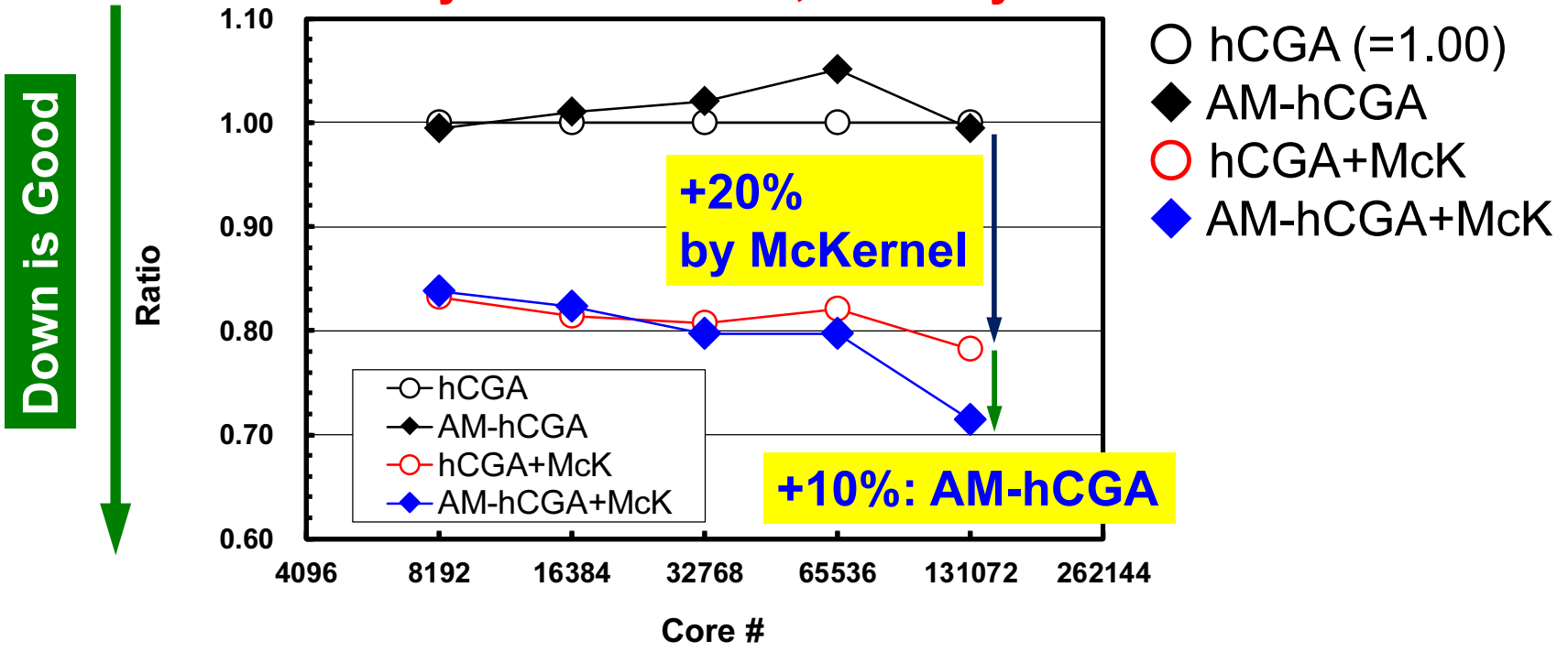


With IHK/McKernel

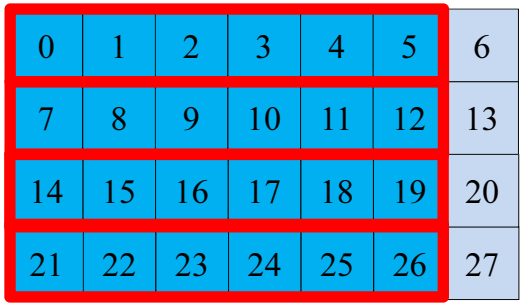


"Tiny" Cases: Significant Improvement by IHK/McKernel and AM-*h*CGA [KN ScalA19]

Computation time is normalized by that of *h*CGA
+20% by IHK/McKernel, +10% by AM-*h*CGA



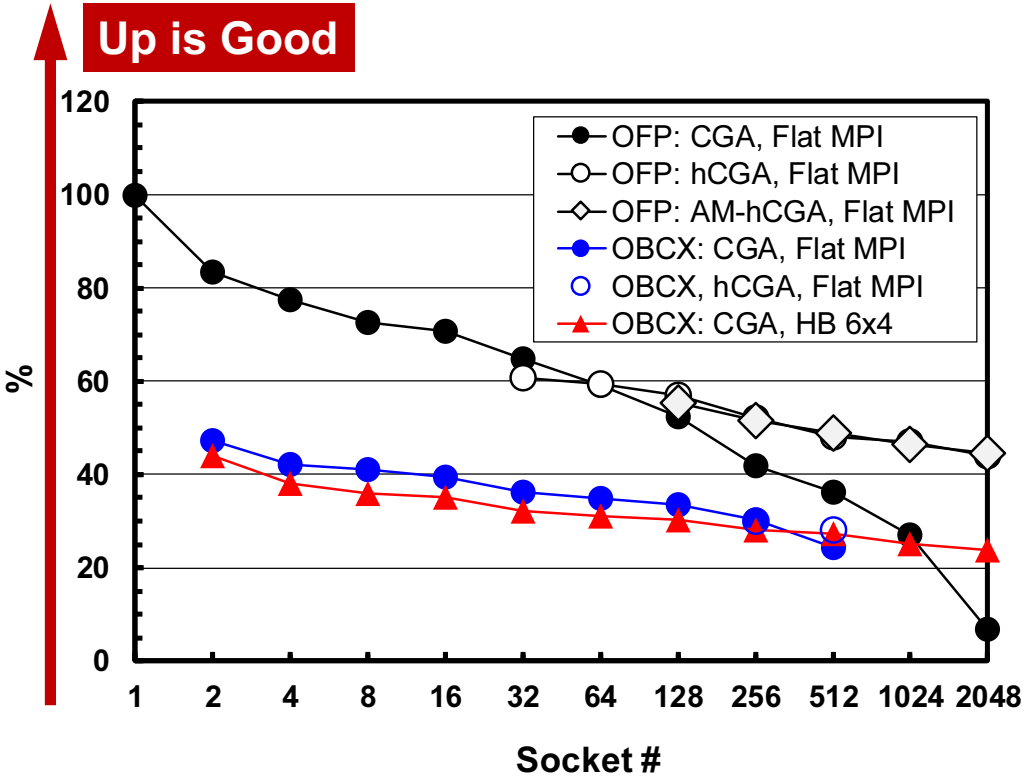
Present Work: Configurations



- OFP
  - Flat MPI
  - Flat Mode/MCDRAM Only
  - IHK/McKernel
- OBCX
  - **Flat MPI (Only 16 of 28 cores are used on each Soc.)**
  - **HB 6x4 (6-threads x 4-proc's per 1 socket (28 cores))**
  - NO IHK/McKernel (Linux Only)
- Weak Scaling
  - 1~2,048 Sockets (1,024 nodes for OBCX)
  - 4,194,304 DOF/Socket (up to 8,589,934,592 DOF): Medium in [KN 2019]
- Strong Scaling
  - 2~2,048 Sockets
  - 134,217,728 DOF ($=512^3$)
- **Time for MGCG solver is evaluated, Best of 5 measurements**

Weak Scaling: Parallel Performance

Performance=100% at 1-node of OFP

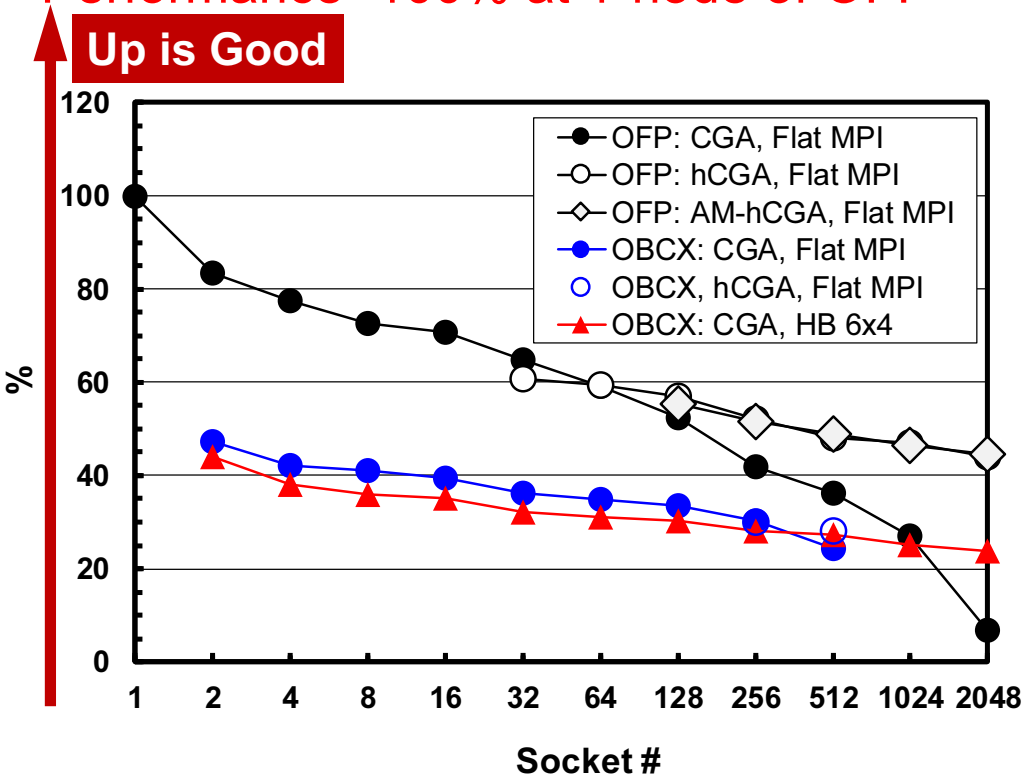


- If number of sockets (MPI processes) increases:
  - ✓ More Iterations
    - ➢ 40@1-Soc.⇒56@2,048-Soc.
  - ✓ More Communication Overhead

Weak Scaling: Parallel Performance

OFP vs. OBCX

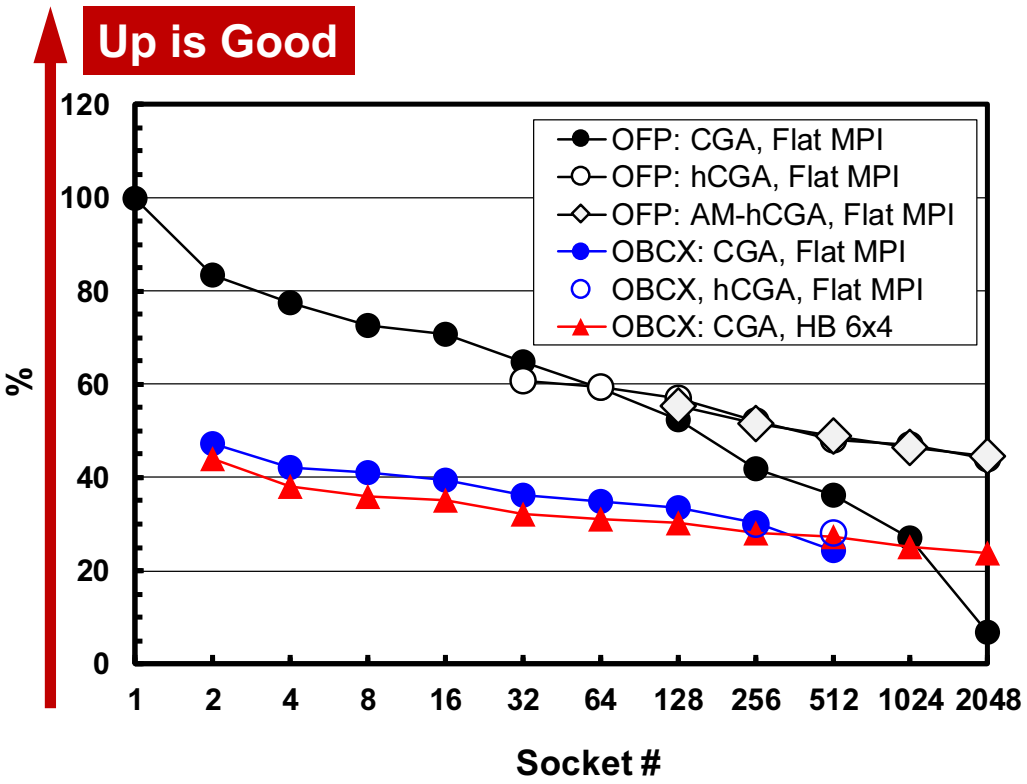
Performance=100% at 1-node of OFP



- Similar FLOPS/Socket
- 4-5x Stream Memory Bandwidth
  - ✓ 490 :101
- MGCG
  - ✓ Sparse Matrices: Memory-Bound
- Actual Performance is 2:1
  - ✓ Actual Throughput of OFP is 200-250 GB/sec if it is not fully vectorized
  - ✓ Reasonable Ratio

Weak Scaling: Parallel Performance

Performance=100% at 1-node of OFP

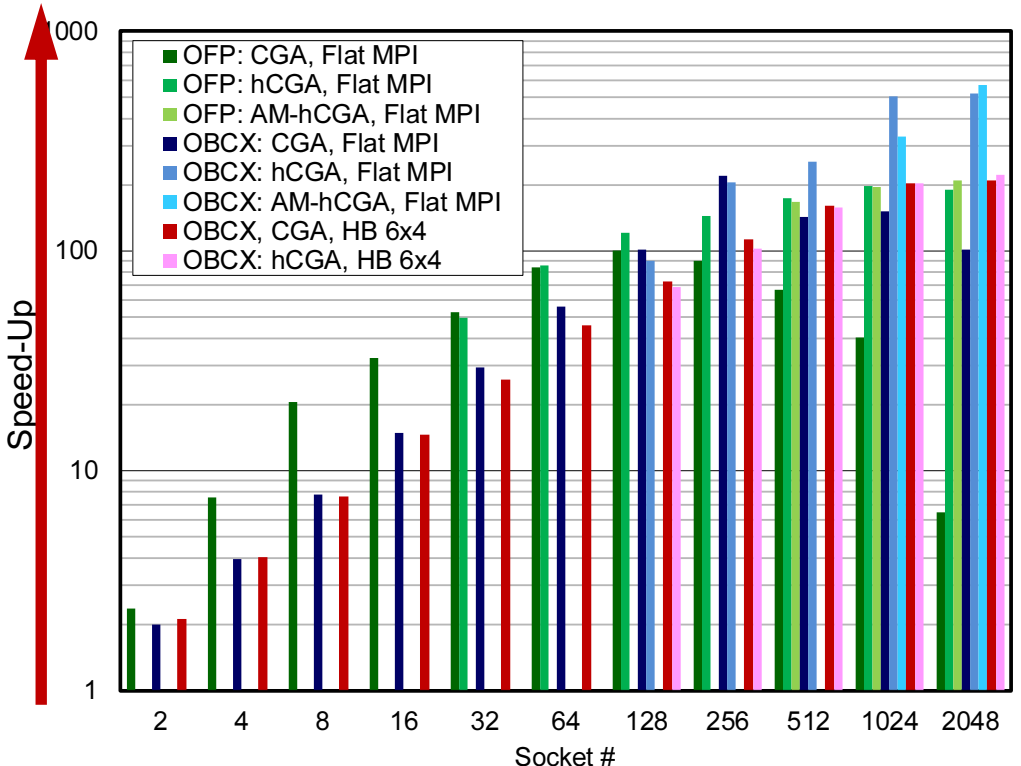


- CGA vs. *h*CGA vs. AM-*h*CGA
  - ✓ OFP
    - ➢ Performance of CGA is getting worse if Soc# is more than 256
    - ➢ hCGA~AM-hCGA for Medium Sized Problems
  - ✓ OBCX
    - ➢ Performance of CGA is not so bad
  - ✓ Flat MPI
    - ➢ Coarse grid solver is on a single core
- Flat MPI vs. HB 6x4 on OBCX
  - ✓ Flat MPI is slightly better, but HB is better if Soc.# is larger.

Strong Scaling: Parallel Performance

Speed-Up= 2.00 for OBCX at 2-Soc's, Flat MPI, CGA (The paper is wrong)

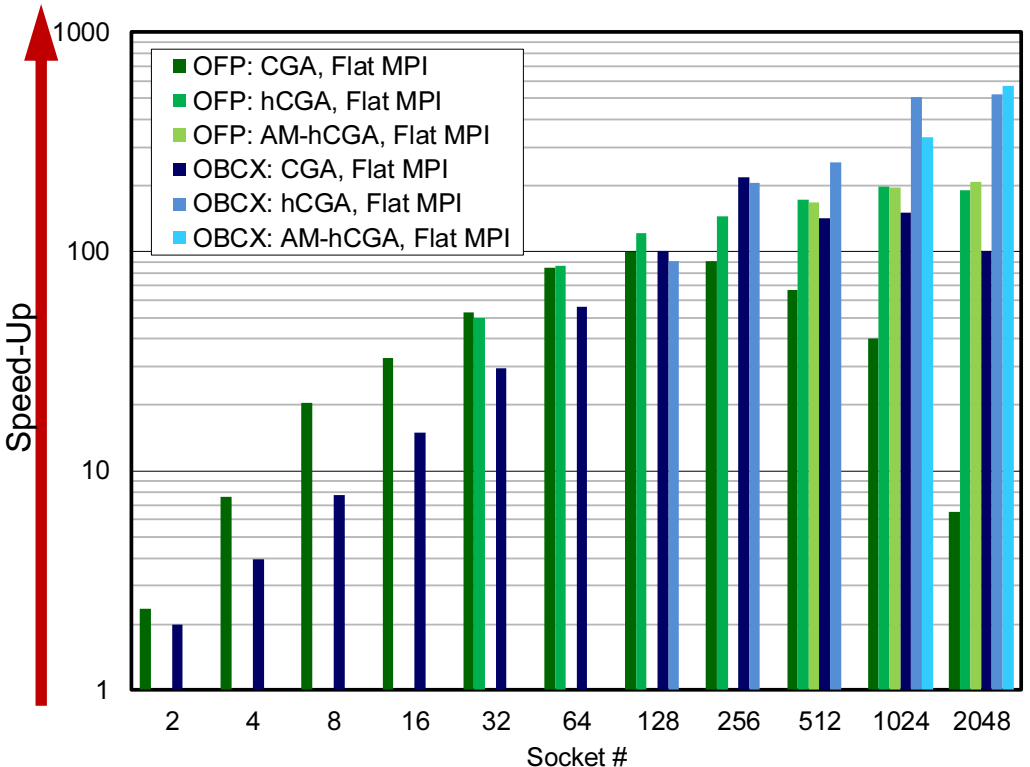
Up is Good



Strong Scaling: Parallel Performance

Speed-Up= 2.00 for OBCX at 2-Soc's, Flat MPI, CGA (The paper is wrong)

Up is Good



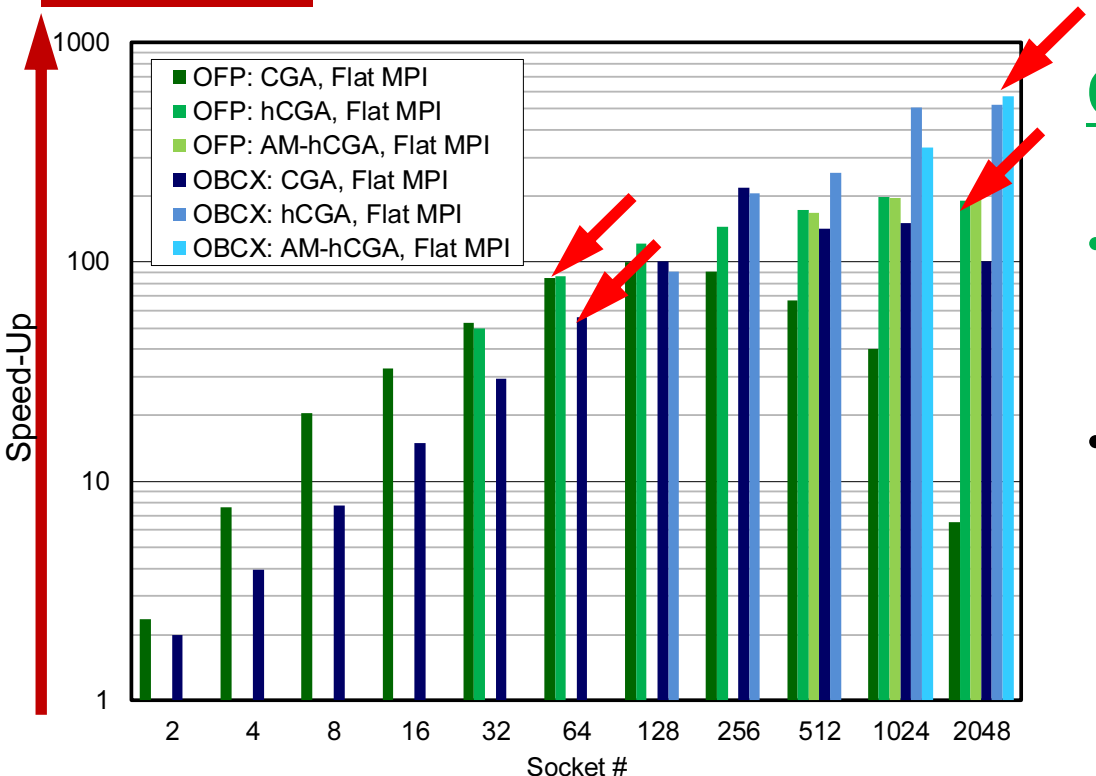
OFP vs. OBCX

- **OFP** is rather faster if Soc.# is smaller, but **OBCX** outperforms at more Soc.#
- Effects of *h*-CGA/AM-*h*CGA is very significant on OBCX with more than 1,024 sockets.

Strong Scaling: Parallel Performance

Speed-Up= 2.00 for OBCX at 2-Soc's, Flat MPI, CGA (The paper is wrong)

Up is Good



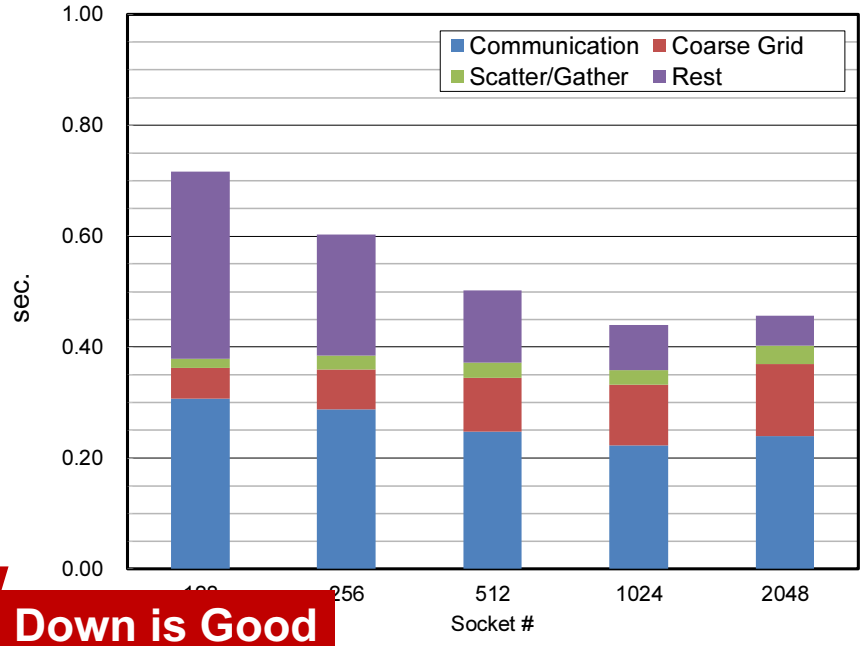
OFP vs. OBCX

- **OFP** is rather faster if Soc.# is smaller, but **OBCX** outperforms at more Soc.#
- Effects of *h*-CGA/AM-*h*CGA is very significant on OBCX with more than 1,024 sockets.

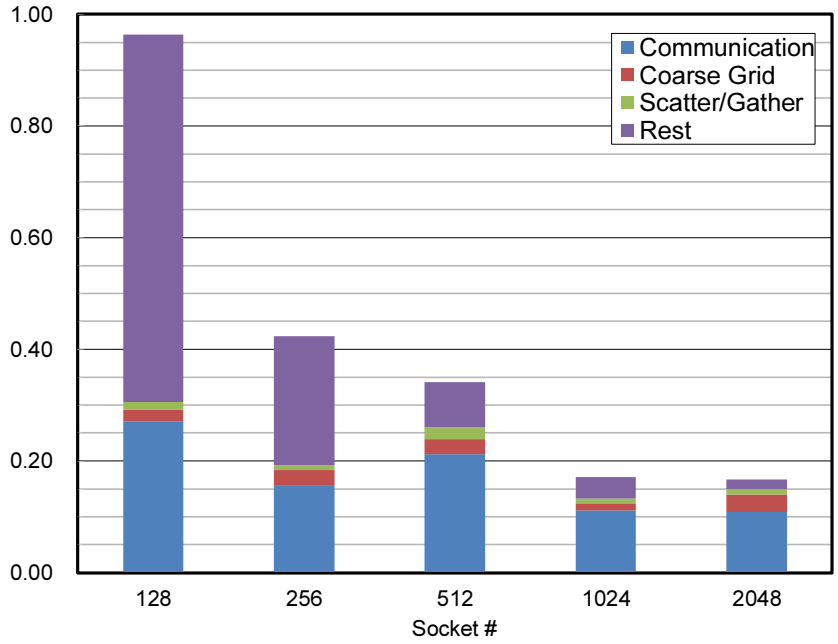
Strong Scaling: Elapsed Time for MGCG Flat MPI, *h*CGA

Send/Recv, Allreduce
Coarse Grid
Scatter/Gather
Rest: Smoother

OFP

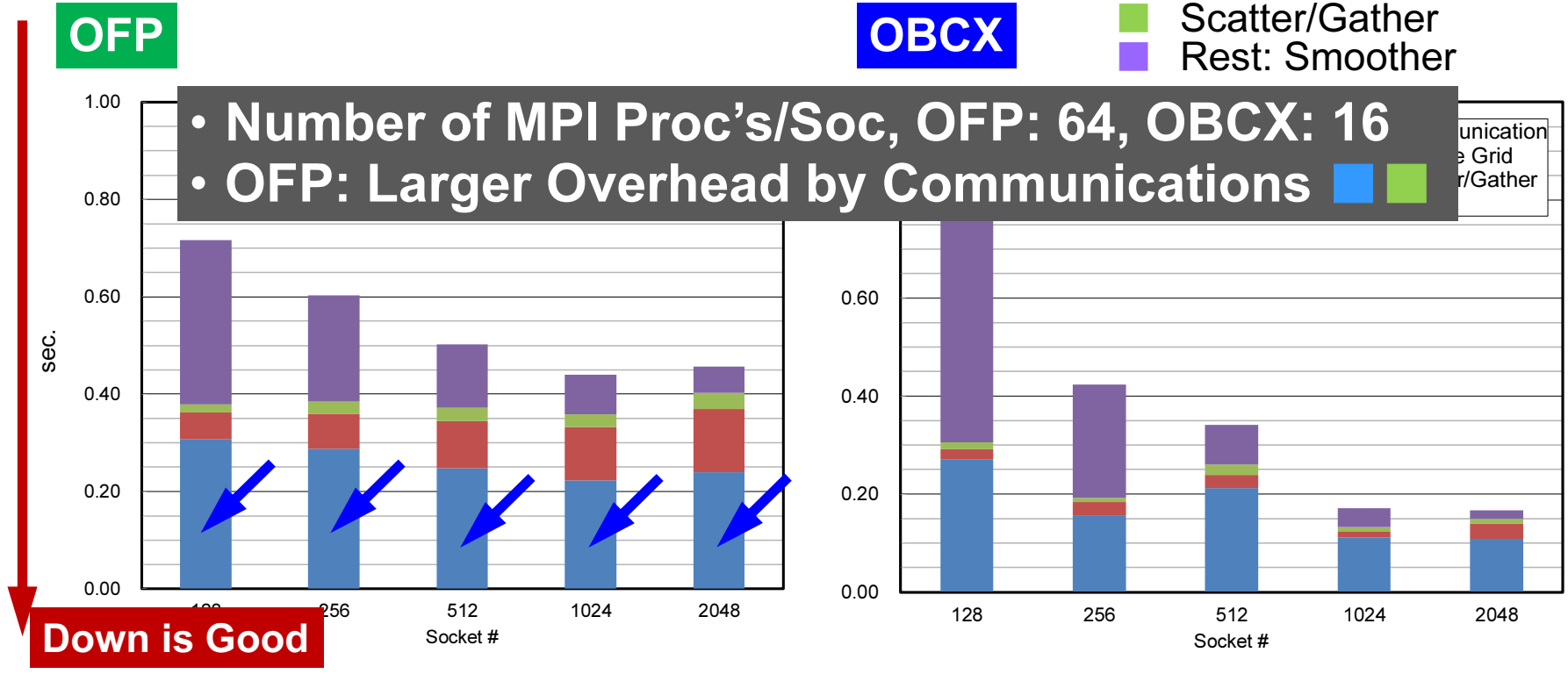


OBCX

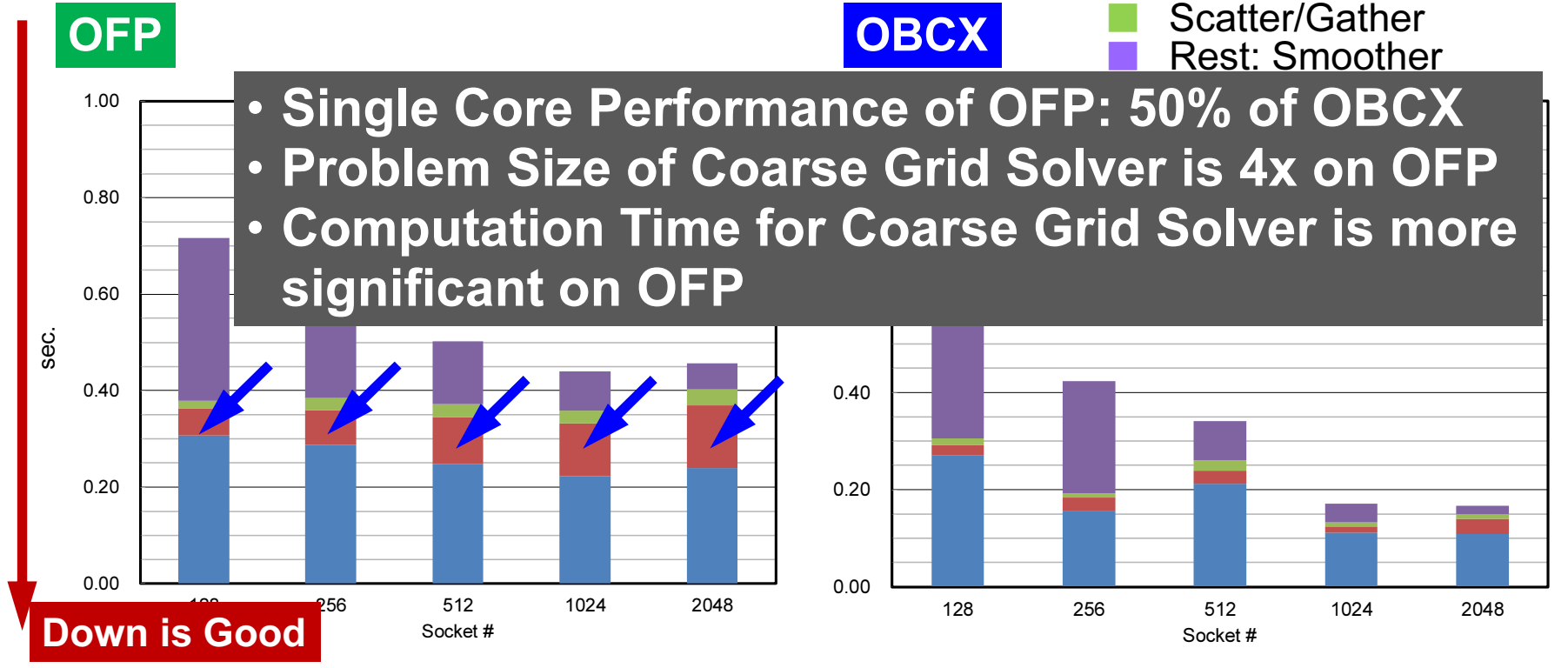


Down is Good

Strong Scaling: Elapsed Time for MGCG Flat MPI, *h*CGA



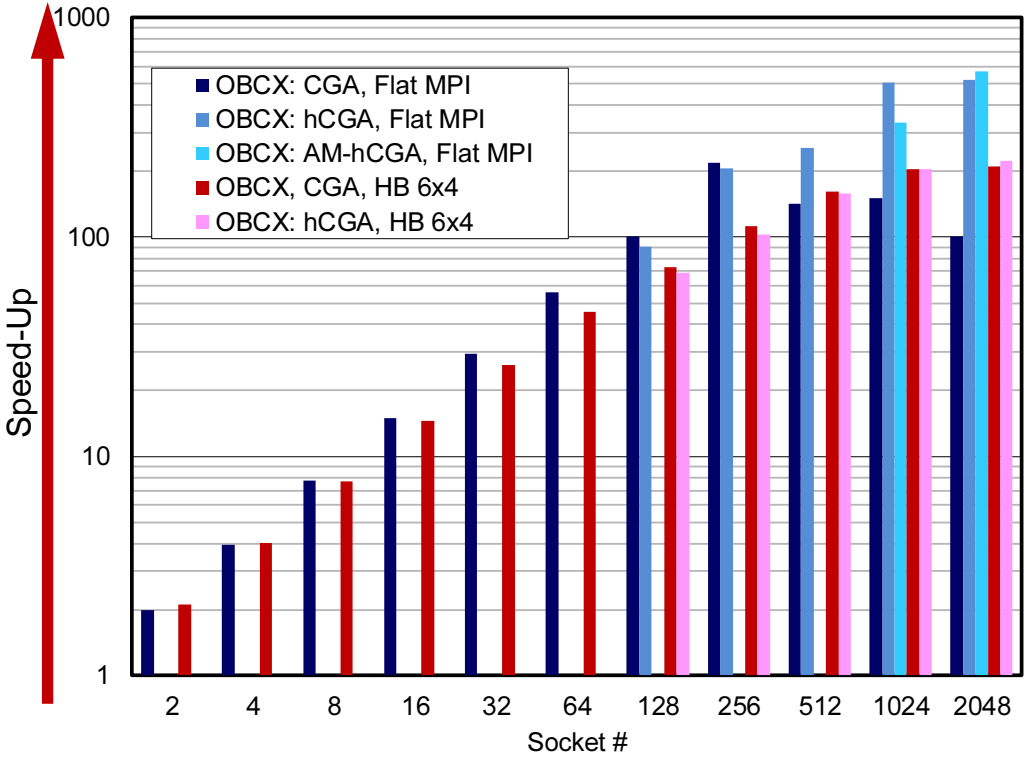
Strong Scaling: Elapsed Time for MGCG Flat MPI, *h*CGA



Strong Scaling: Parallel Performance

Speed-Up= 2.00 for OBCX at 2-Soc's, Flat MPI, CGA (The paper is wrong)

Up is Good



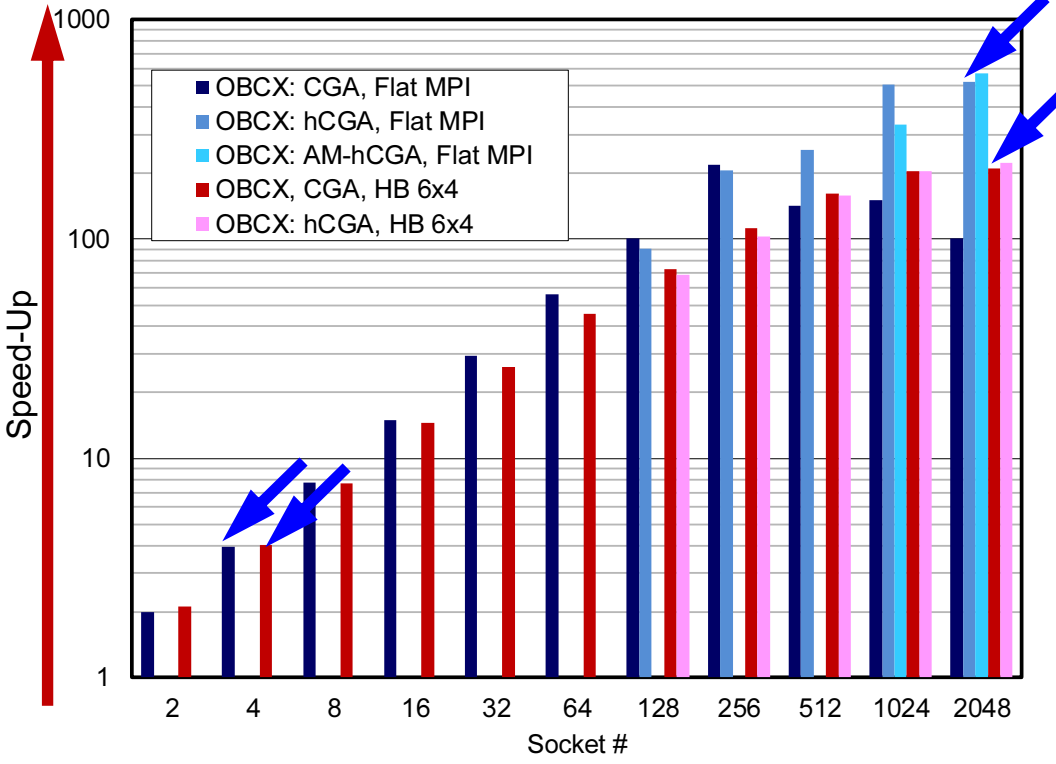
Flat MPI vs. HB 6x4 on OBCX

- **HB 6x4** is generally much slower than **Flat MPI**, especially at more than 1,024 sockets
- Different behavior compared to weak scaling, other systems (Fujitsu FX10)
- Investigations needed

Strong Scaling: Parallel Performance

Speed-Up= 2.00 for OBCX at 2-Soc's, Flat MPI, CGA (The paper is wrong)

Up is Good



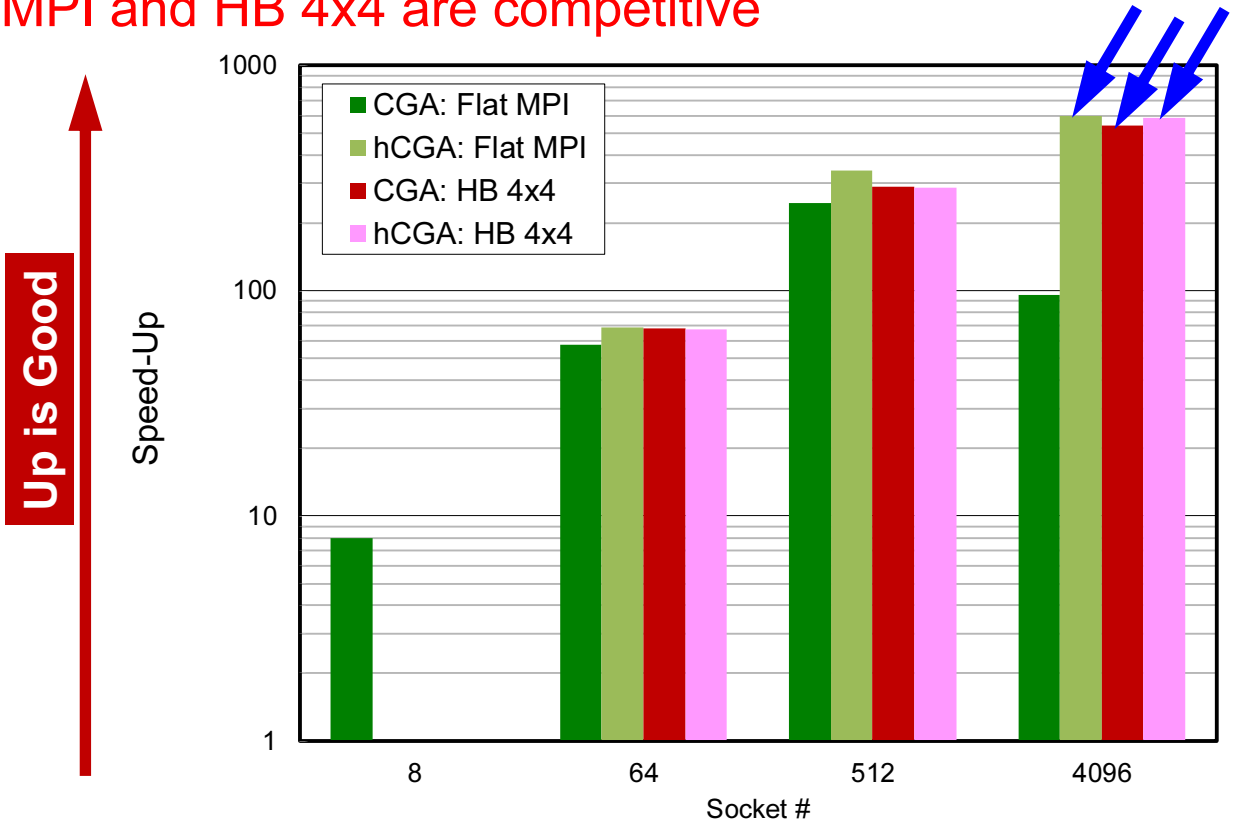
Flat MPI vs. HB 6x4 on OBCX

- **HB 6x4** is generally much slower than **Flat MPI**, especially at more than 1,024 sockets
- Different behavior compared to weak scaling, other systems (Fujitsu FX10)
- Investigations needed

Strong Scaling: Parallel Performance [KN 2014]

Speed-Up= 8.00 for FX10 at 2-Nodes, Flat MPI, CGA

Flat MPI and HB 4x4 are competitive



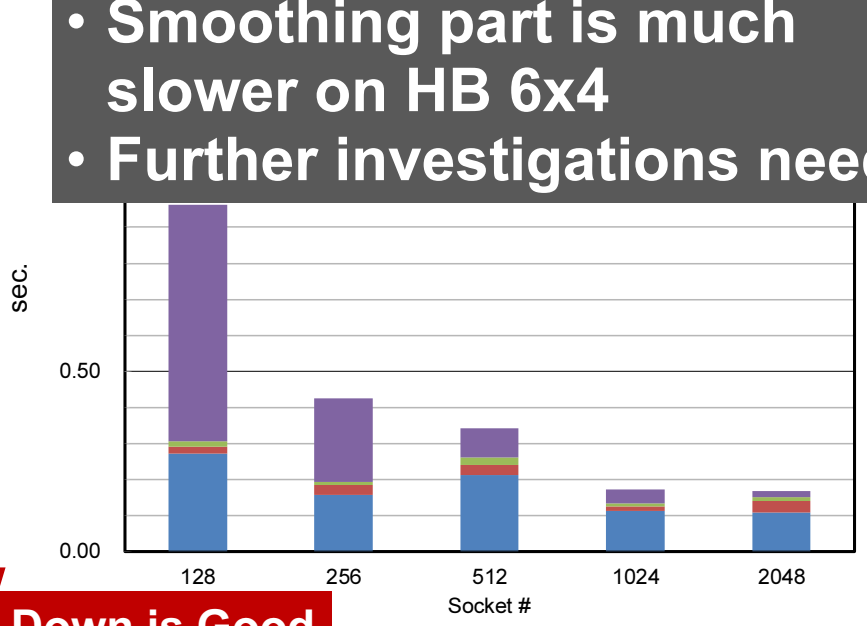
Strong Scaling: Elapsed Time for MGCG OBCX, *h*CGA

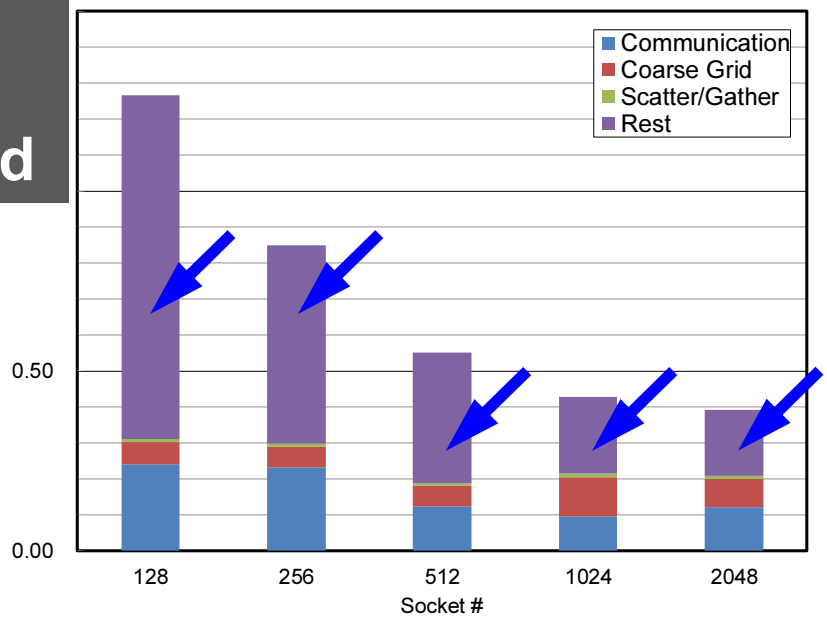
Send/Recv, Allreduce
Coarse Grid
Scatter/Gather
Rest: Smoother

Flat MPI

Hybrid 6x4

- Smoothing part is much slower on HB 6x4
- Further investigations needed





Down is Good

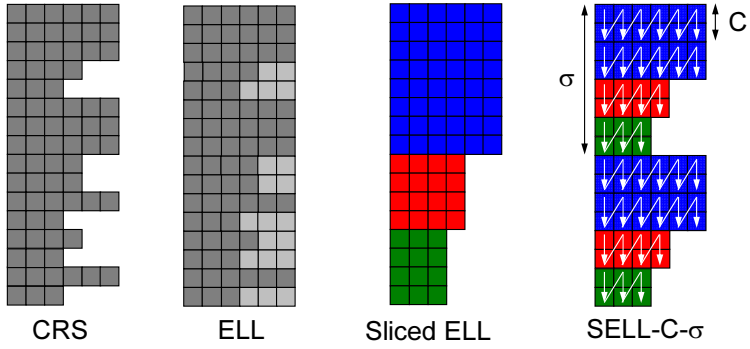
Conclusions (1/2)

- The *h*CGA and the AM-*h*CGA provide excellent performance on both of OFP and OBCX with large number of nodes.
- In weak scaling, performance of OFP is generally twice as much as that of OBCX.
- Although OFP is faster than OBCX for smaller number of sockets (no more than 128) in strong scaling, OBCX outperforms with more than 256 sockets.
  - MPI proc. # on OFP (=64) is four times as large as that on OBCX (=16)
  - Problem size for the coarse grid solver on OFP is also four times larger, and coarse grid solver is executed on a single core for *Flat MPI*
  - Performance of a single core of OFP is half of OBCX.

Conclusions (2/2)

- Behaviors of OpenMP/MPI Hybrid Parallel Programing Model on OBCX are very different from those on Fujitsu PRIMEHPC FX10 used in the author's previous work.
  - Generally speaking, coarse grid solver, SpMV, and smoothing operators in HB 6×4 are more expensive than those in Flat MPI on OBCX.
  - Further investigations and optimizations for these procedures in multithreading is needed
- Generally, overhead of multithreading by OpenMP on manycore architectures, such as KNL, is significant.
  - Special treatment for do/for-loops parallelized by OpenMP proposed in [8] will be applied to OpenMP/MPI hybrid version of the code.
- Currently, IHK/McKernel is not available on OBCX
  - Installation and evaluation of IHK/McKernel is also expected in near future.
  - Fluctuations of comp. time is not so significant on OBCX (5-10% at 2,048 sockets)

More Future Works



- Pipelined Algorithms
- SELL-C-σ
- Lower/Mixed Precision
- Preliminary Results: Double/Single Precision
  - Number of Iterations does not change
  - Computation Time for MGCG
    - 0.85 for OFP (Single/Double Ratio)
    - 0.60 for Intel BDW Cluster (Single/Double Ratio)
  - Further Vectorization Needed on OFP
- Larger Problems using More Cores