SAND2015-2707C

Parallel Deterministic Transport Sweeps of Structured and Unstructured Meshes with Overloaded Mesh Decompositions

Shawn D. Pautz
Sandia National Laboratories
Albuquerque, NM

Teresa Bailey
Lawrence Livermore National Laboratory
Livermore, CA

ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method

Nashville, TN
April 19-23, 2015

SCEPTRE

Sandia National Laboratories

Exceptional service in the national interest

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

SAND2015-??? PE

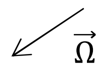
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Outline

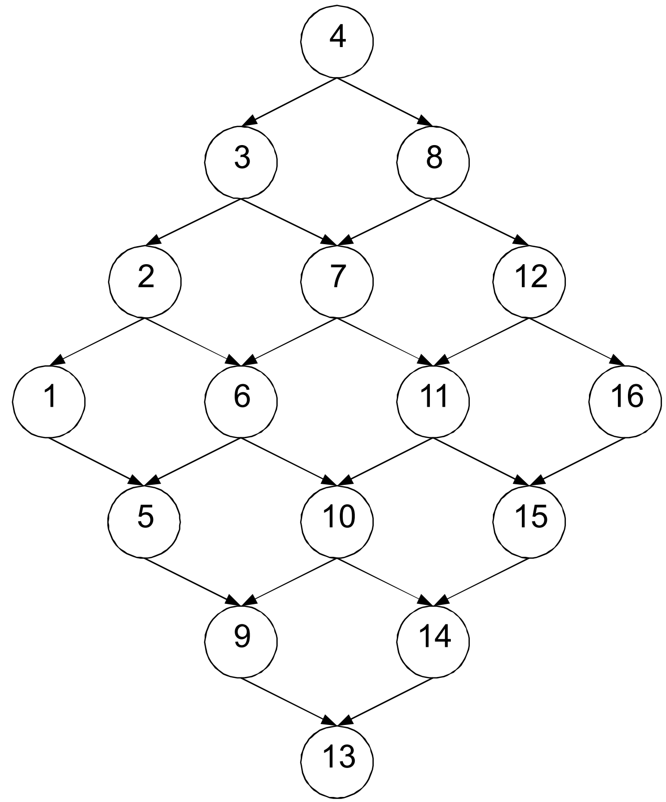
- Challenge for parallel scaling of sweeps
- General characterization of sweep algorithms
- Review of overloading technique for structured meshes
- Extension of structured overloading technique to unstructured meshes
- Results
- Conclusions

Challenge for parallel sweep scaling

Sandia National Laboratories



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



Amount of concurrency < number of tasks

Challenge is to exploit existing concurrency

Characterization of sweep algorithms

Parallel sweep algorithms may be described by their approach to several issues

- Mesh partitioning
- Task aggregation
- Communication aggregation
- Task scheduling

Note that some approaches have used mesh partitioning to define task aggregation. Also, task and communication aggregation have typically been considered the same concept.

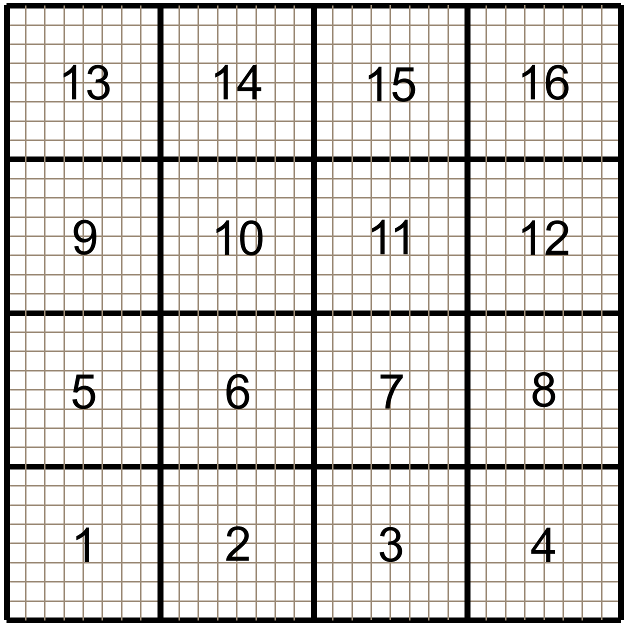
Structured vs. unstructured algorithms

Various parallel sweep techniques have been originally conceived and analyzed for structured meshes. Extending them to the unstructured case is often difficult.

We have recently extended the method of mesh overloading to the unstructured case.

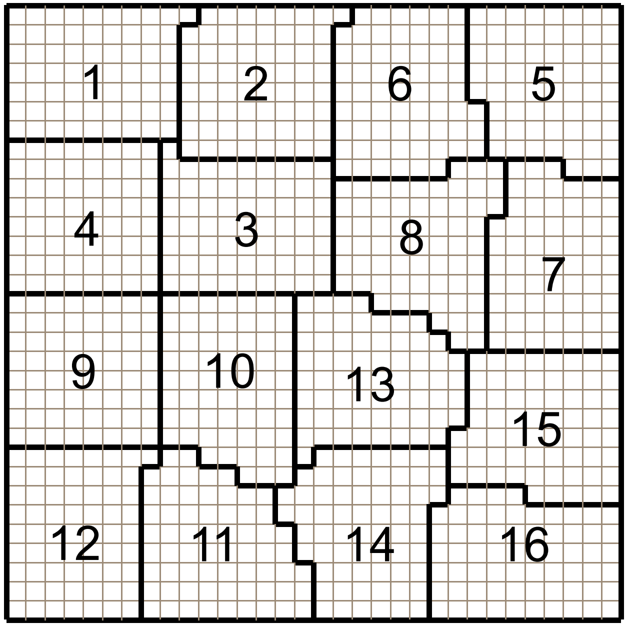
Note: “unstructured” algorithms may be applied to structured meshes, which is useful for comparison to structured algorithms.

Standard structured mesh partitioning



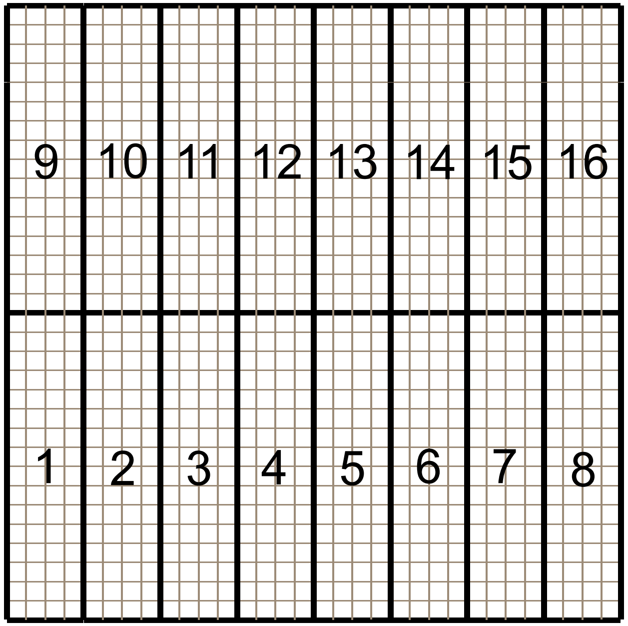
The standard approach to structured mesh partitioning subdivides the mesh into a $P_x x P_y$ set of subdomains. It suffers from poor parallel sweep scaling.

Standard unstructured mesh partitioning



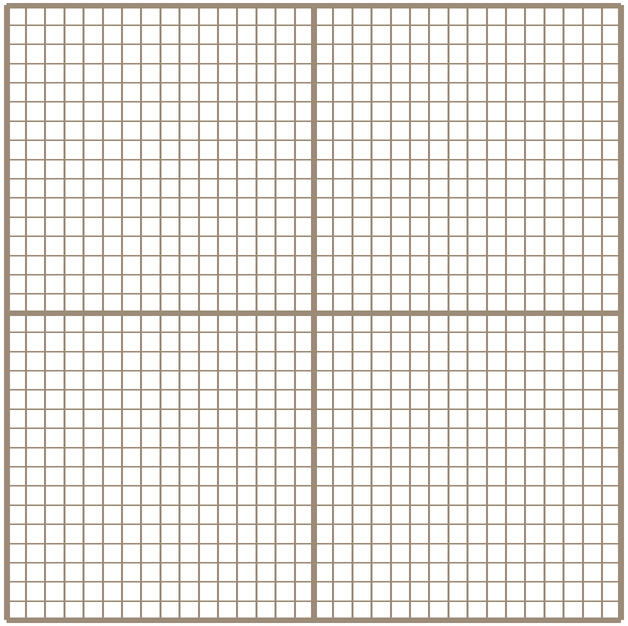
The standard approach to unstructured mesh partitioning subdivides the mesh into a set of P compact subdomains (e.g. with Chaco, Metis), which is a generalization of the structured approach. It also suffers from poor parallel sweep scaling.

Hybrid KBA partitioning



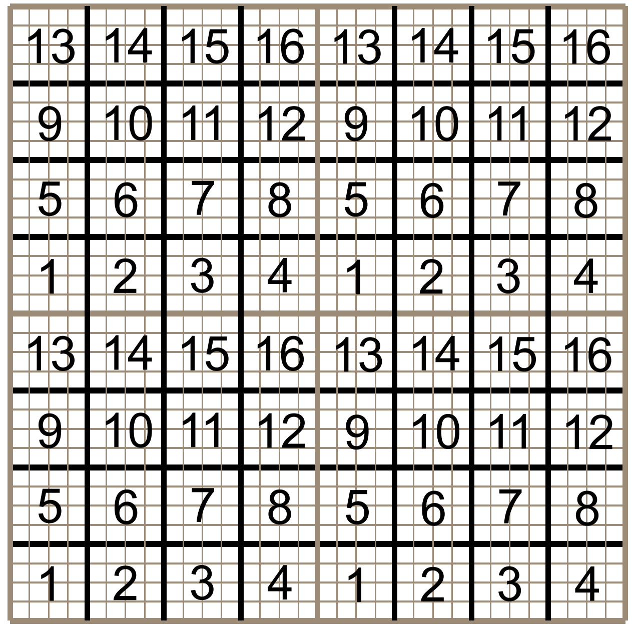
The hybrid KBA approach to structured mesh partitioning subdivides the mesh into a P_xx2 set of subdomains. It generally gives good parallel sweep scaling, but it is not clear how to extend it to unstructured meshes.

Structured mesh overloading



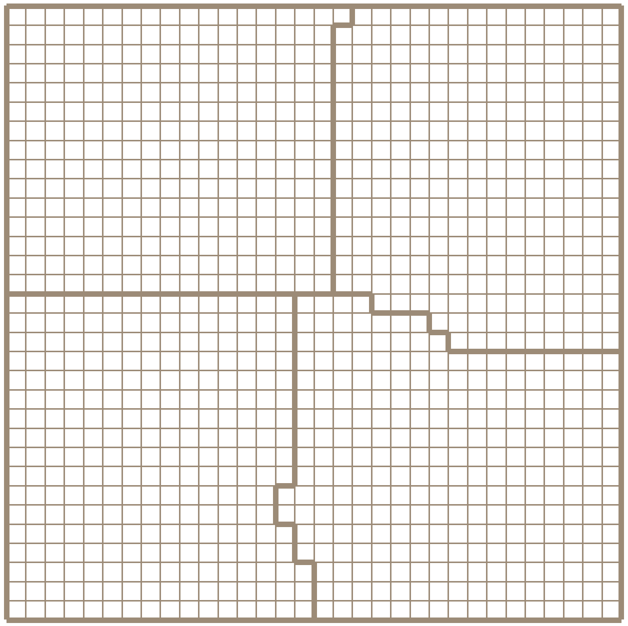
In the overloading approach for structured meshes, we first (conceptually) divide the mesh into a set of $\omega_x x \omega_y$ “tiles”.

Structured mesh overloading, continued



Next we divide each tile into a set of $P_x x P_y$ subdomains. Each process gets a subdomain from each tile. This increases the sweep concurrency, but it also increases communication costs.

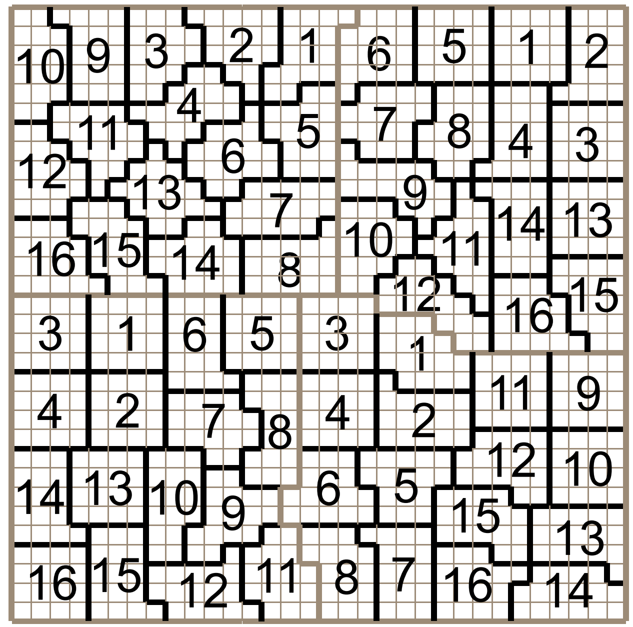
Unstructured mesh overloading



We extend the overloading approach to unstructured meshes by first dividing the mesh into a set of ω tiles by means of the usual mesh/graph partitioning algorithms.

Unstructured mesh overloading, continued

Sandia National Laboratories



Next we divide each tile into a set of P subdomains. Each process gets a subdomain from each tile. This increases the sweep concurrency, but it also increases communication costs.

Advantages of unstructured mesh overloading

Sandia National Laboratories

- Concepts from structured algorithms may be generalized to unstructured meshes
- May reuse existing algorithms created for standard unstructured mesh partitioning
- May balance computational and communication costs by means of the overloading parameter ω

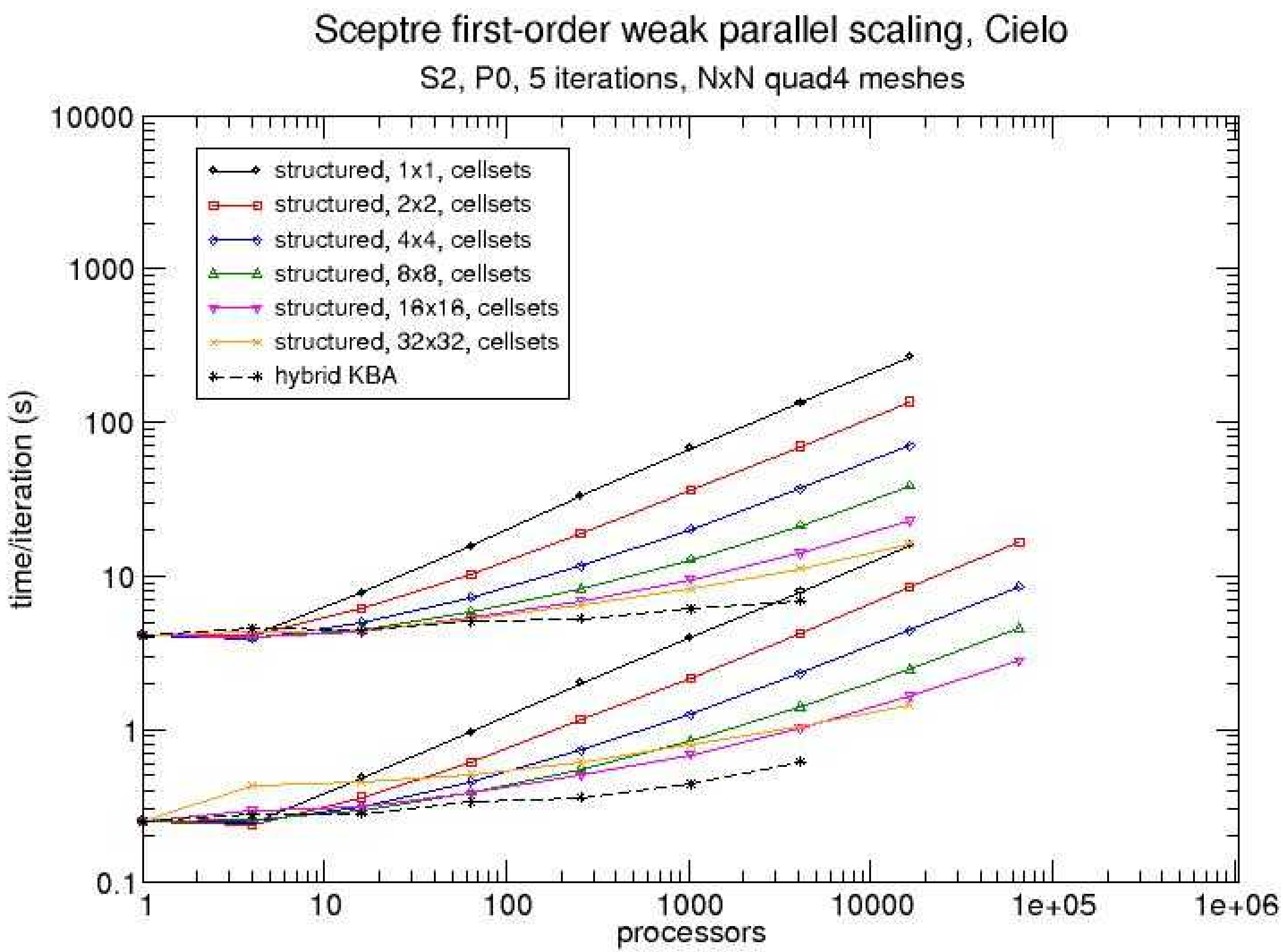
Results

The Sceptre code was used to perform studies of the overloading approach for both structured and unstructured algorithms, in both cases applied to structured meshes. Sceptre itself is a general unstructured mesh code; it has no knowledge of structured properties. Partitioning occurs outside of the code as a preprocessing step.

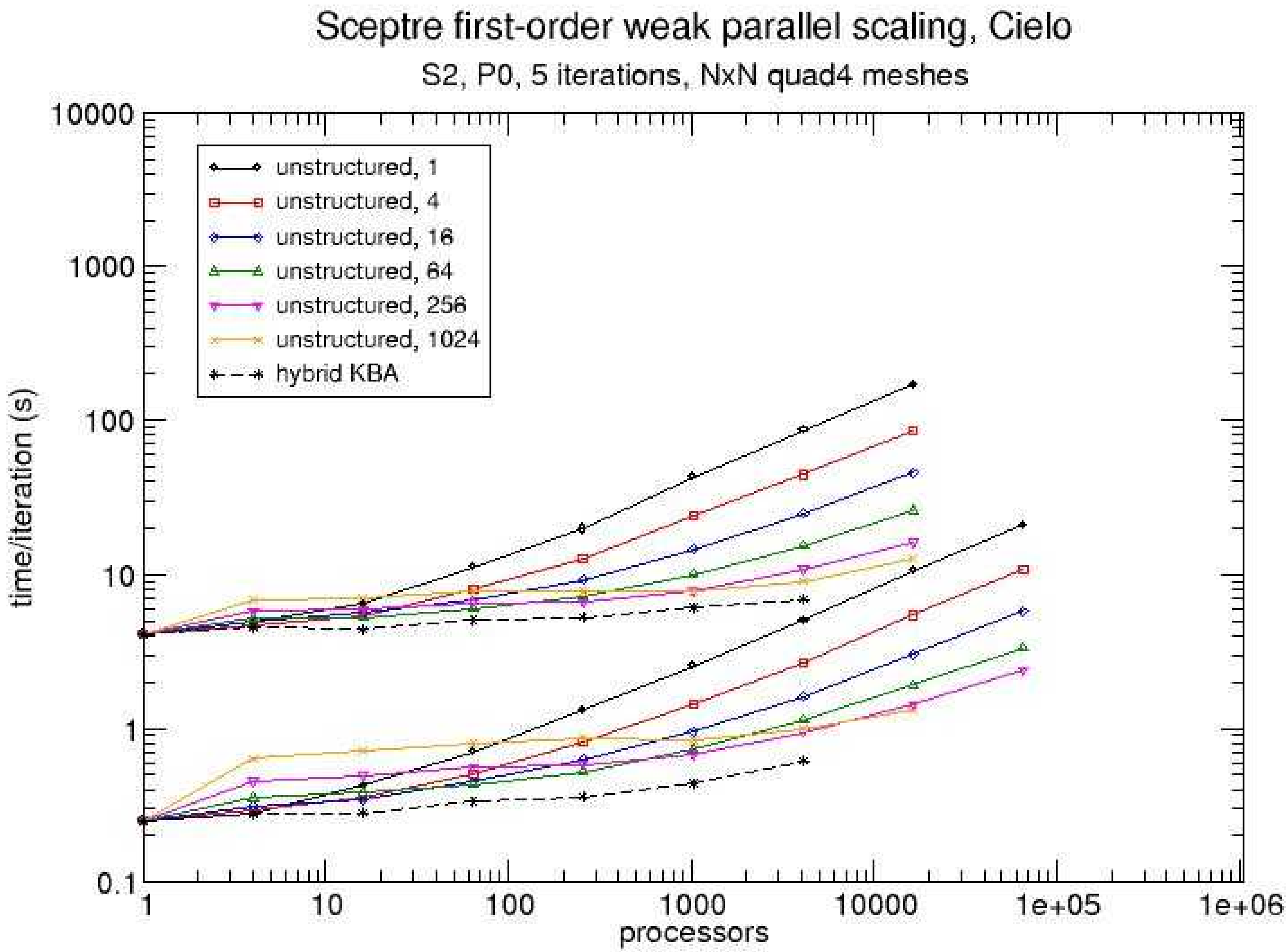
For the structured case we use cellsets that are identical to a full subdomain within a tile. For the unstructured case we use individual cells as a cellset. Communication occurs after the completion of each cellset.

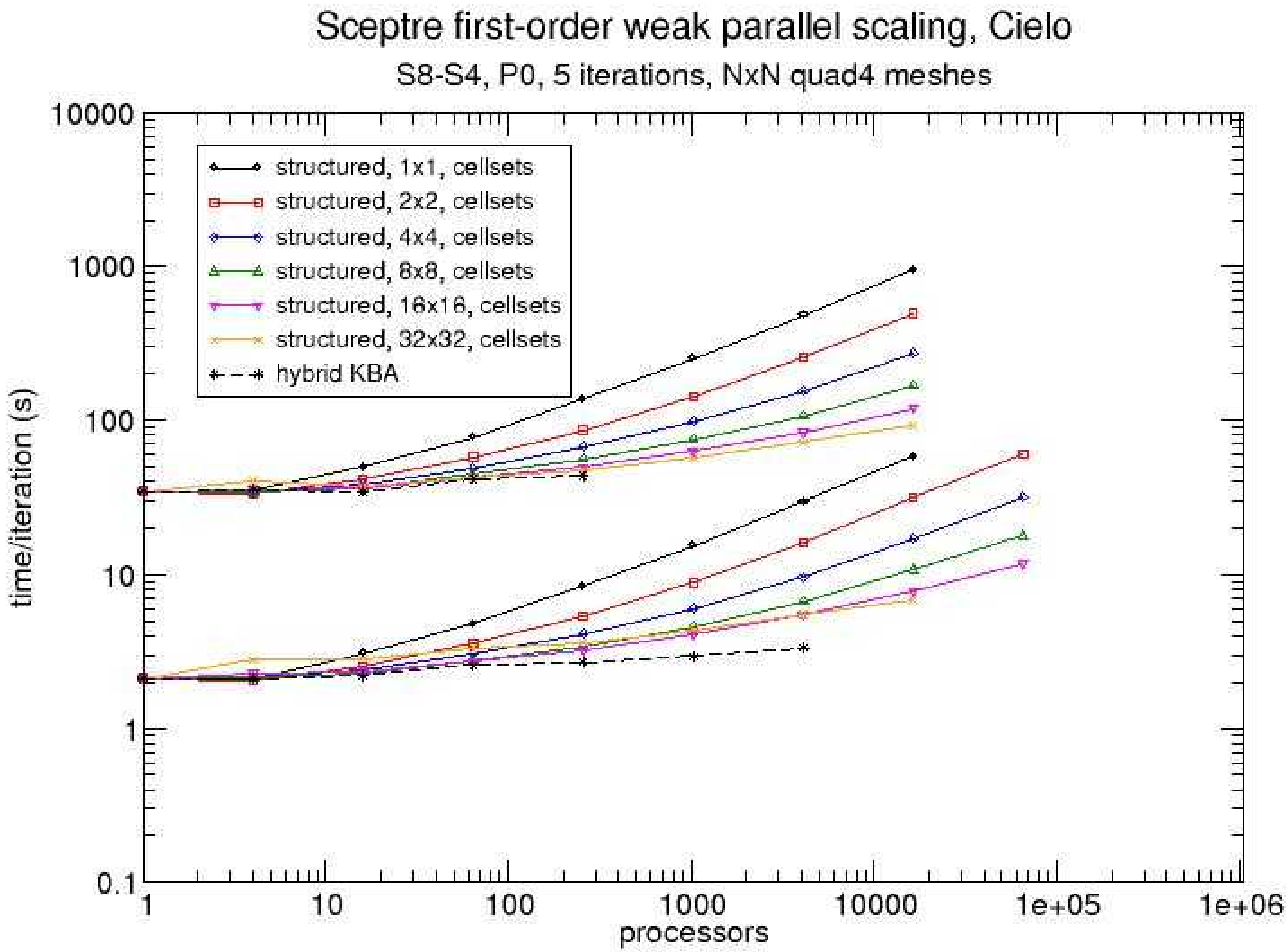
We use the DFHDS scheduling heuristic in all cases.

All results were generated on Cielo, a massively parallel computer at Los Alamos National Laboratory.

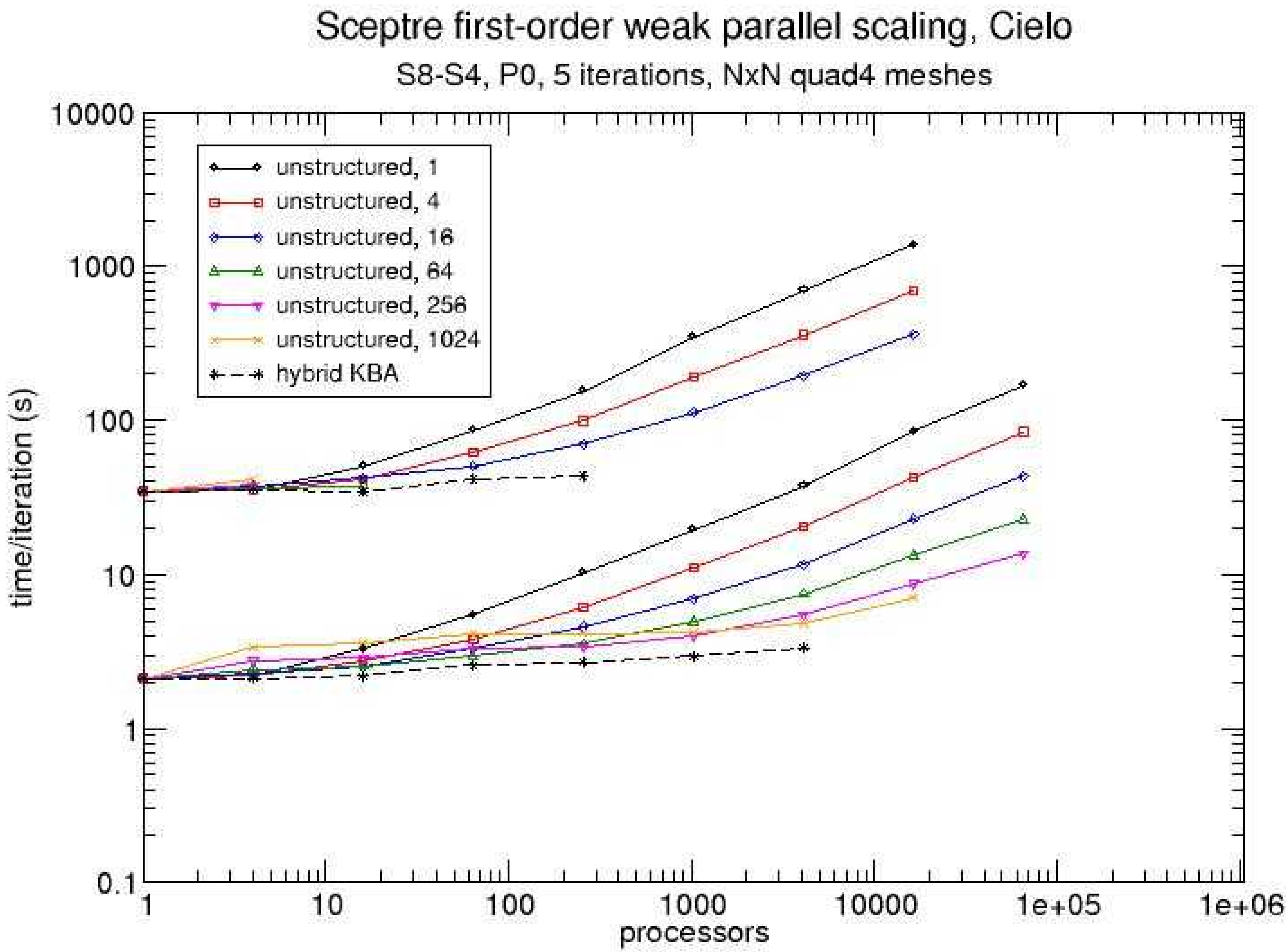


Sceptre first-order weak parallel scaling, Cielo
S2, P0, 5 iterations, NxN quad4 meshes
structured, 1x1, cellsets
structured, 2x2, cellsets
structured, 4x4, cellsets
structured, 8x8, cellsets
structured, 16x16, cellsets
structured, 32x32, cellsets
hybrid KBA
time/iteration (s)
processors
10000
1000
100
10
1
0.1
1
10
100
1000
10000
1e+05
1e+06

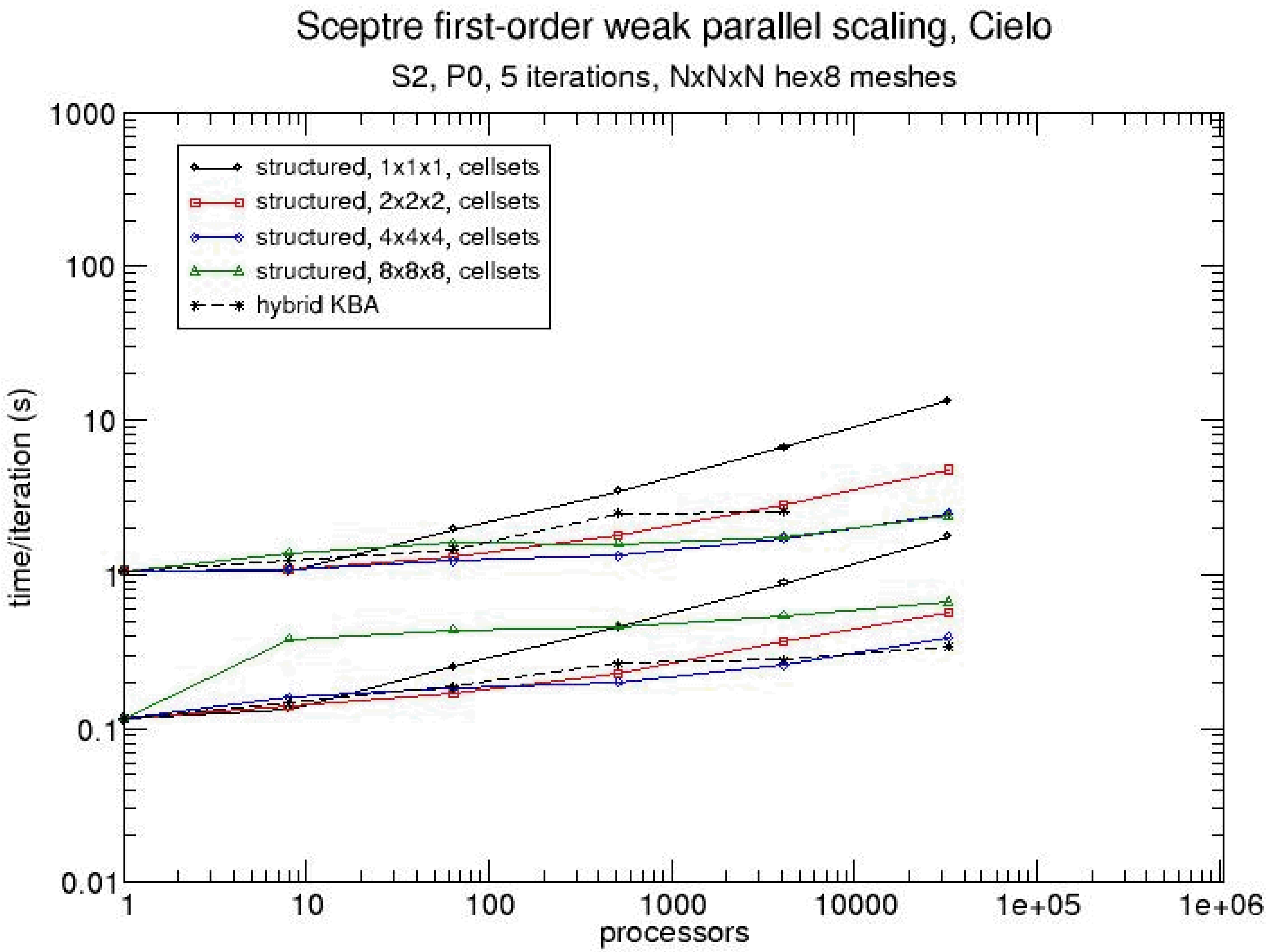

Sceptre first-order weak parallel scaling, Cielo
S2, P0, 5 iterations, NxN quad4 meshes
unstructured, 1
unstructured, 4
unstructured, 16
unstructured, 64
unstructured, 256
unstructured, 1024
hybrid KBA
time/iteration (s)
processors
10000
1000
100
10
1
0.1
1
10
100
1000
10000
1e+05
1e+06



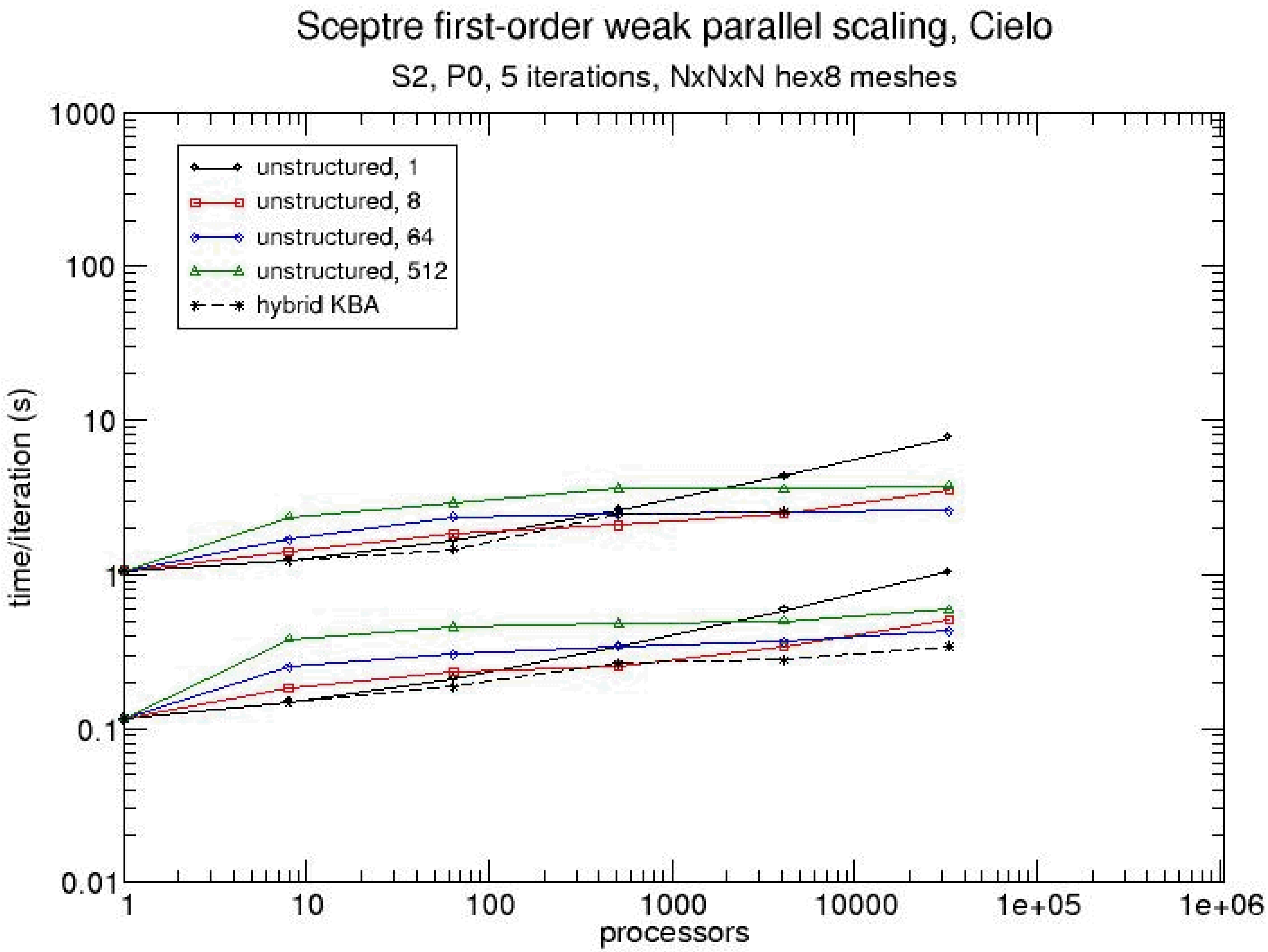
Sceptre first-order weak parallel scaling, Cielo
S8-S4, P0, 5 iterations, NxN quad4 meshes
structured, 1x1, cellsets
structured, 2x2, cellsets
structured, 4x4, cellsets
structured, 8x8, cellsets
structured, 16x16, cellsets
structured, 32x32, cellsets
hybrid KBA
time/iteration (s)
10000
1000
100
10
1
0.1
1
10
100
1000
10000
1e+05
1e+06
processors

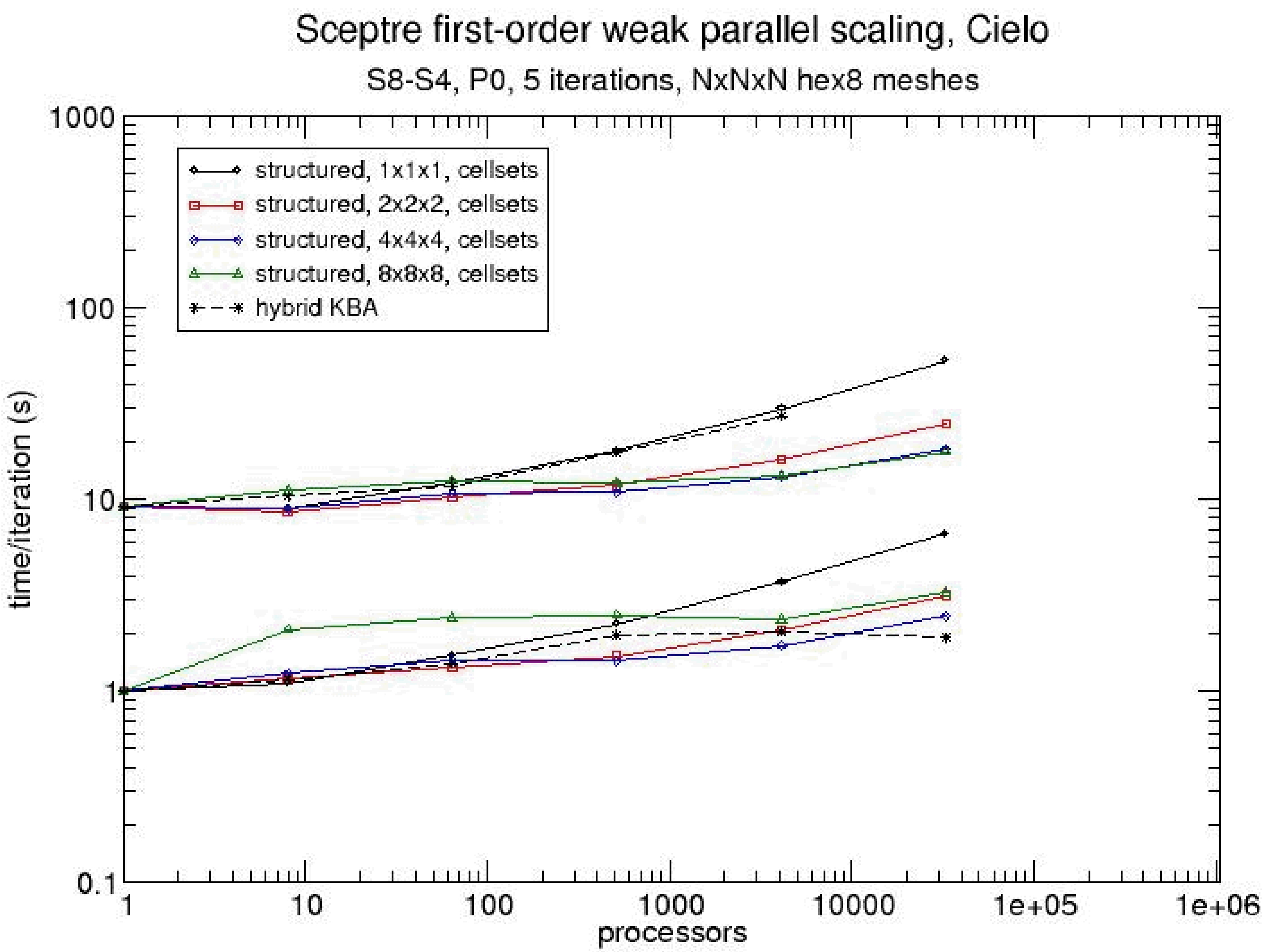


Sceptre first-order weak parallel scaling, Cielo
S8-S4, P0, 5 iterations, NxN quad4 meshes
unstructured, 1
unstructured, 4
unstructured, 16
unstructured, 64
unstructured, 256
unstructured, 1024
hybrid KBA
time/iteration (s)
processors
10000
1000
100
10
1
0.1
1
10
100
1000
10000
1e+05
1e+06

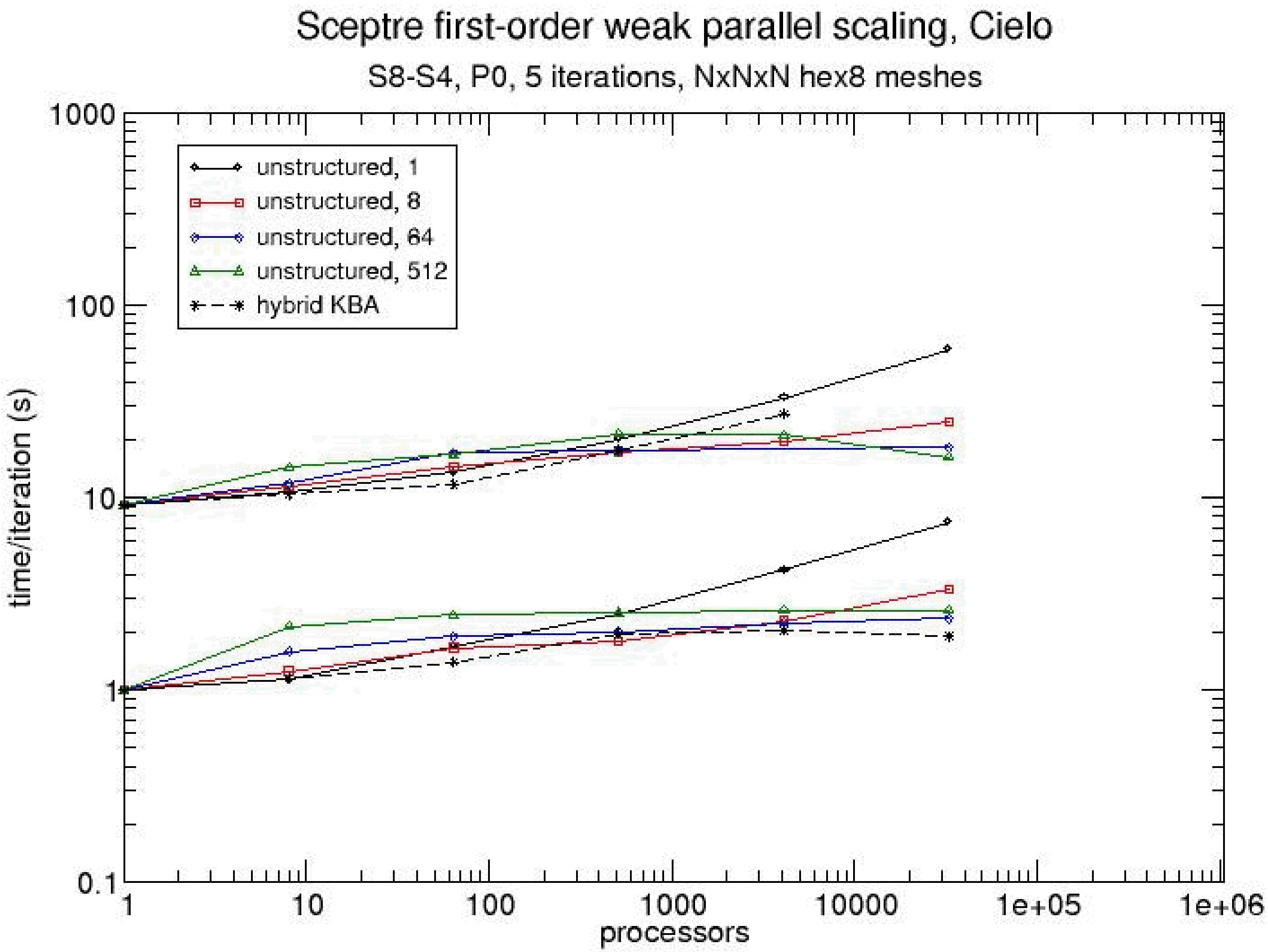


Sceptre first-order weak parallel scaling, Cielo
S2, P0, 5 iterations, NxNxN hex8 meshes
structured, 1x1x1, cellsets
structured, 2x2x2, cellsets
structured, 4x4x4, cellsets
structured, 8x8x8, cellsets
hybrid KBA
time/iteration (s)
1000
100
10
1
0.1
0.01
1
10
100
1000
10000
1e+05
1e+06
processors


Sceptre first-order weak parallel scaling, Cielo
S2, P0, 5 iterations, NxNxN hex8 meshes
unstructured, 1
unstructured, 8
unstructured, 64
unstructured, 512
hybrid KBA
time/iteration (s)
processors
1000
100
10
1
0.1
0.01
1
10
100
1000
10000
1e+05
1e+06



Sceptre first-order weak parallel scaling, Cielo
S8-S4, P0, 5 iterations, NxNxN hex8 meshes
structured, 1x1x1, cellsets
structured, 2x2x2, cellsets
structured, 4x4x4, cellsets
structured, 8x8x8, cellsets
hybrid KBA
time/iteration (s)
processors
1000
100
10
1
0.1
1
10
100
1000
10000
1e+05
1e+06


Sceptre first-order weak parallel scaling, Cielo
S8-S4, P0, 5 iterations, NxNxN hex8 meshes
1000
100
10
1
0.1
time/iteration (s)
1
10
100
1000
10000
1e+05
1e+06
processors
unstructured, 1
unstructured, 8
unstructured, 64
unstructured, 512
hybrid KBA

Conclusions

Sandia National Laboratories

- We have successfully extended the concept of overloading to the general case
- The technique leverages existing tools
- The technique leads to improved scaling for both structured and unstructured algorithms
- More studies are desired to consider other aggregation and scheduling heuristics
- Challenge: performance models for the general case
  - Unstructured vs. structured
  - More general aggregation approaches