Optimizing Gauss-Seidel Smoother in HPCG

Jongsoo Park and
Mikhail Smelyanskiy
Parallel Computing Lab, Intel

Notice and Disclaimers

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information. Contact your local Intel sales office or your distributor to obtain the latest specification before placing your product order.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All products, dates, and figures are preliminary for planning purposes and are subject to change without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined.“ Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The Intel products discussed herein may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's website at http://www.intel.com.

Intel® Itanium®, Intel® Xeon®, Xeon Phi™, Pentium®, Intel SpeedStep® and Intel NetBurst® , Intel®, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Copyright © 2012, Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others..

Notice and Disclaimers Continued …

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

Single MPI Performance

Xeon E5-2697 v2

1skt 12-core running at 2.7 GHz, 50 GB/s stream BW

160^3 input (performance changes little for bigger inputs)

SpMVM:

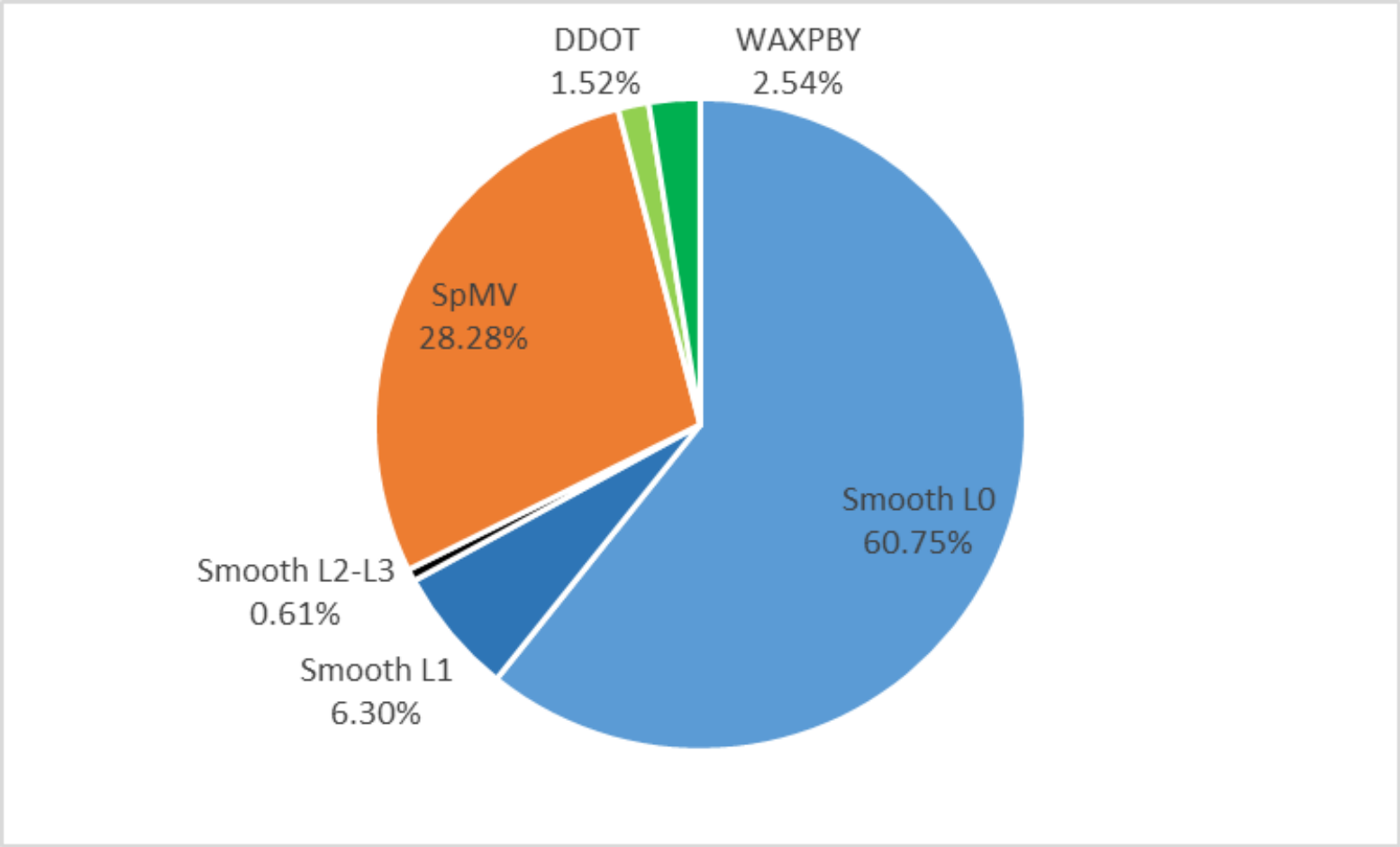
~ 8GFLOP/s or 48 GB/s (96% efficiency)

Smoother for the finest level of MG

~ 7GFLOP/s or 42 GB/s (84 % efficiency)

Overall 6.2 GFLOP/s including optimization phase overhead

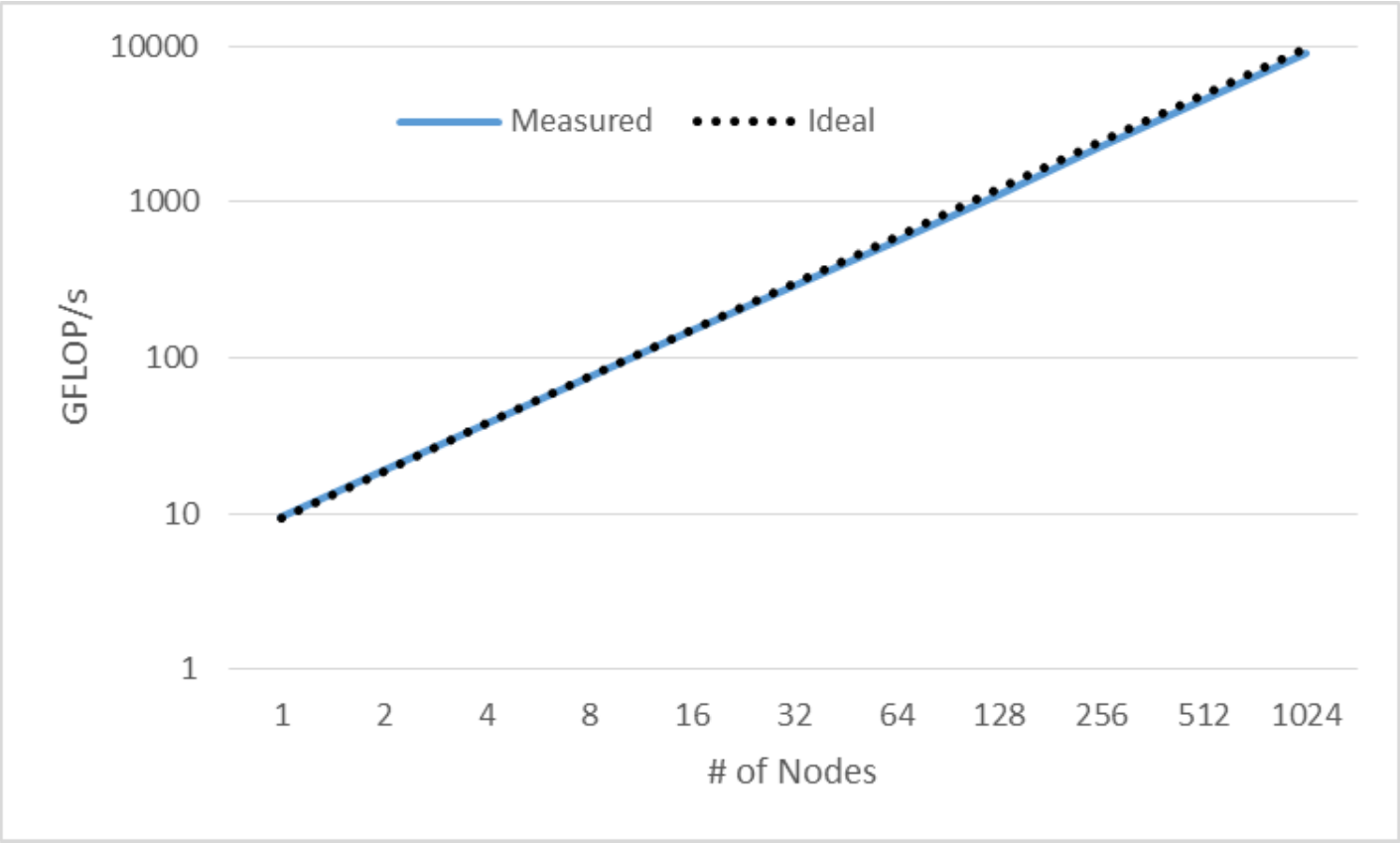
Time Breakdown



Smooth 68%

5

Multi-node Scaling



Stampede cluster, dual socket of 8-core SNB, 2.7 GHz

2 MPI processes per node (1 MPI process per skt. for NUMA)

160^3 input per MPI process

93% parallelization efficiency with 1024 nodes

The Key Optimization Target: Gauss Seidel Smoother

~2/3 of FLOPs in HPCG 2.1

84% of stream BW

- Without reordering: Drops SpMV performance. Often degrades convergence. E.g., 50 → 65 iterations for the same residual, 160^3 input
- 1 MPI process per socket: more MPI processes degrades convergence

Has the same dependency btw rows as sparse triangular solver

7

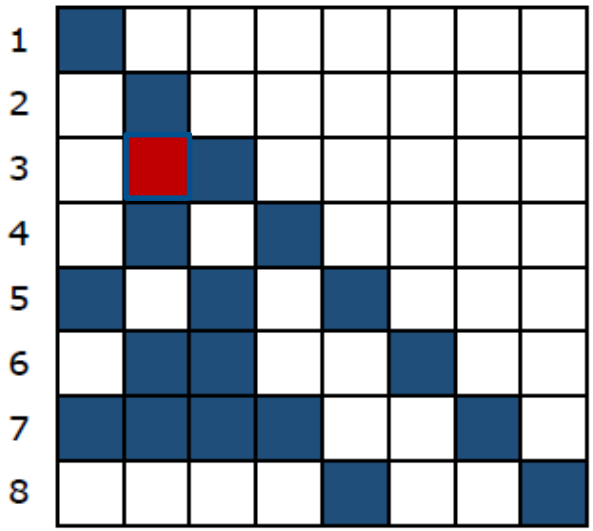
Evaluated sparse matrices

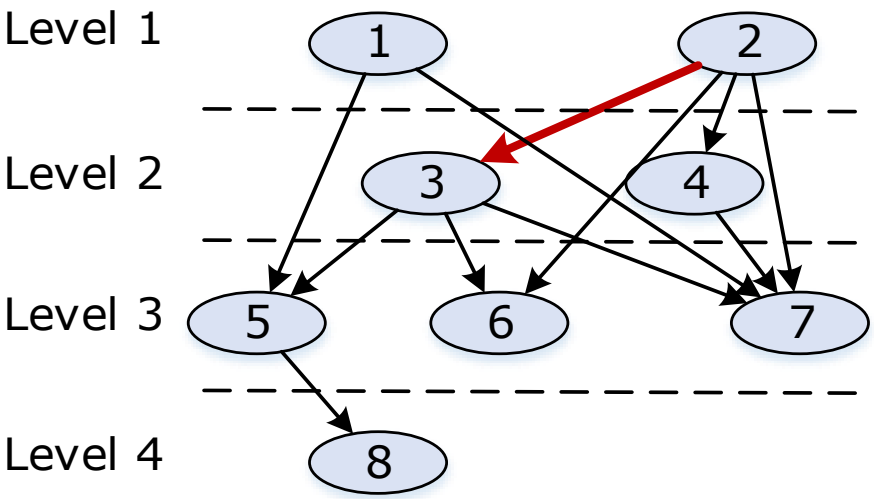
	parallelism	nnz/row
hpcg160	3,657	27
parabolic_fem	75,118	7
apache2	1,077	7
thermal2	991	5
G3_circuit	611	5
ecology2	500	14
StocF-1465	488	73
inline_1	288	44
Geo_1438	247	78
F1		
bmwcra_1		
Emilia_923	176	44
Fault_639	143	45
af_shell3	136	35
Hook_1498	96	41
offshore	75	16
af_3_k101	74	35
BenElechi1	43	53
shipsec8	37	58
ship_003	28	66
crankseg_2	15	222
crankseg_1	13	201

Optimization for a wider class of matrices

U of Florida matrix collection

Task dependency graph of sparse triangular solver





9

Levels in task dependency graph

The level of a node = the longest path length from an entry

A simple way to quantify parallelism

(# of rows)/(# of levels)

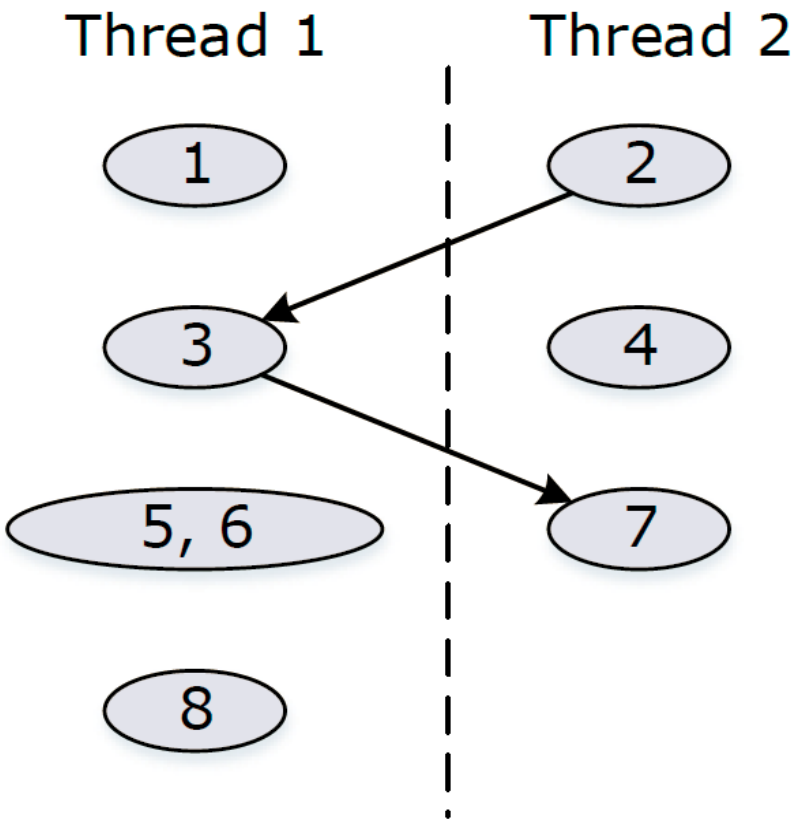
assume row-wise parallelization, same nnz per row

N^3 HPCG problem has ~7N levels

→ $N^2/7$ parallelism

10

Task coarsening and synchronization sparsification



1 super-task per level per thread

Eliminate redundant edges

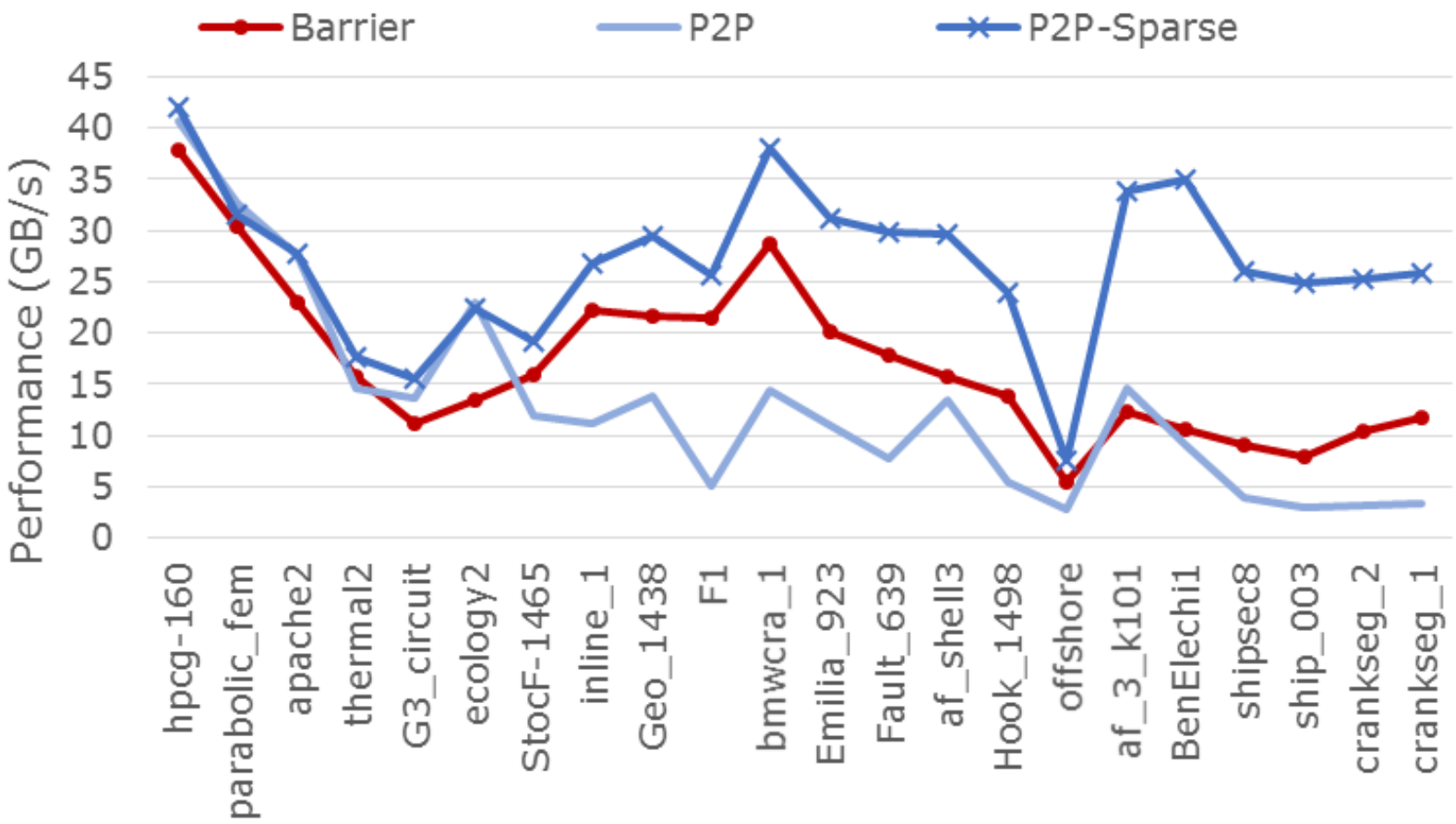
- Duplicated edges
- Intra-thread edges
- Transitive edges

HPCG: about 2.5 synchronizations per level per thread

[Accepted to ISC'14]

11

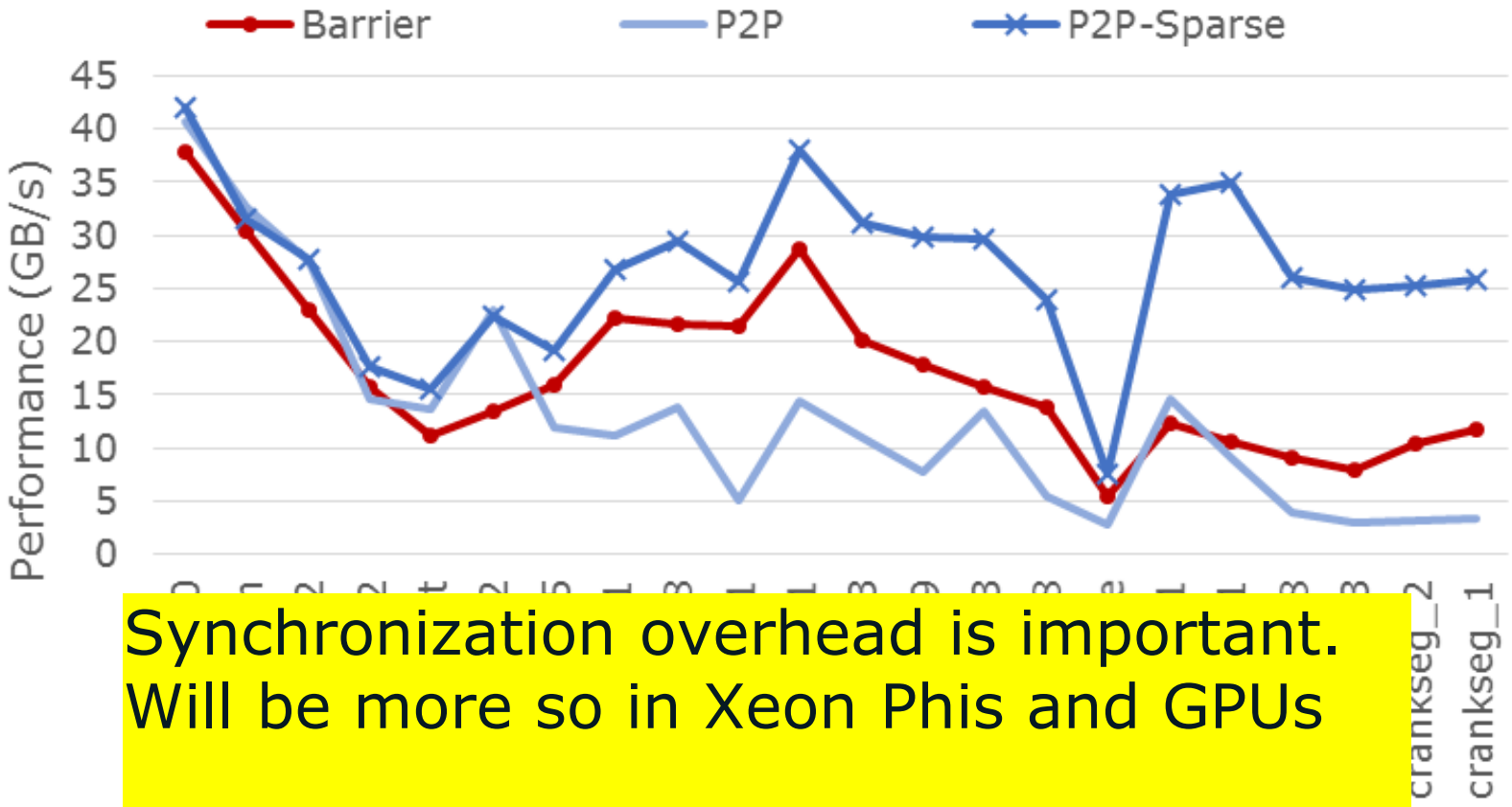
Triangular Solver Performance



12-core machine with 50 GB/s stream BW

12

Triangular Solver Performance



Synchronization overhead is important. Will be more so in Xeon Phis and GPUs

Motivates architecture with fast core-to-core communication

12-cor

13

Summary

Non-trivial optimization efforts required for performance close to stream BW in Xeon

Benefits wider problems: unstructured matrices with limited parallelism, other DAG scheduling problems

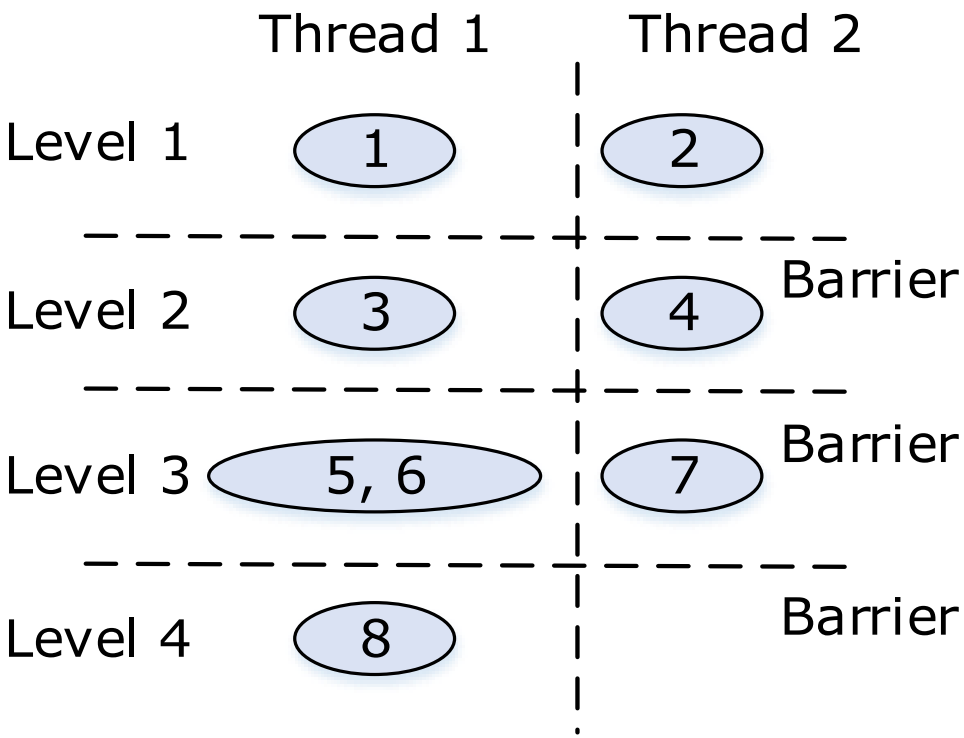
Xeon Phi and GPUs will have more challenges but solving them will have a great impact

14

Backup Slides

15

Level scheduling with barrier synchronization



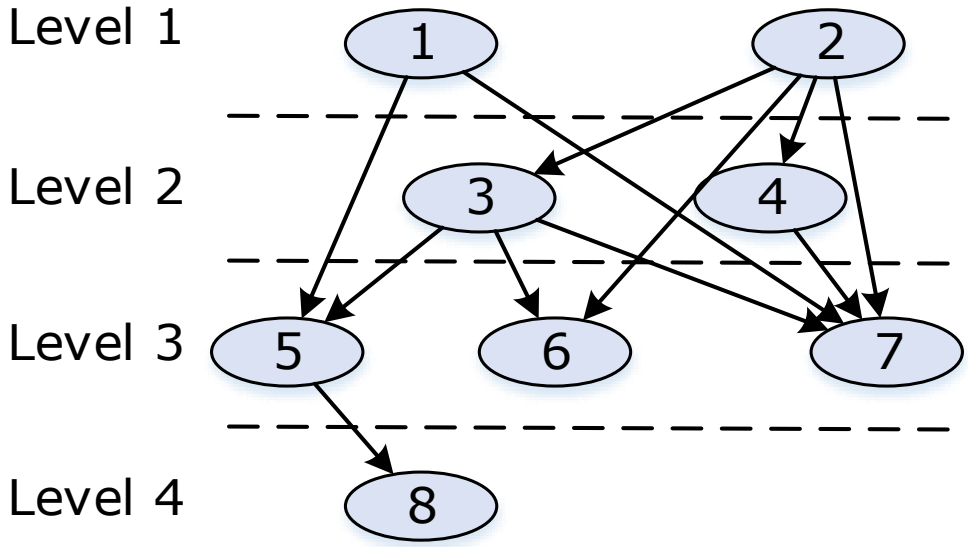
Turn task parallelism into data parallelism (fits GPU well)

Barrier overhead: a few thousand cycles in Xeon (increases with cores)

Per level load imbalance

16

Point-to-point Synchronization



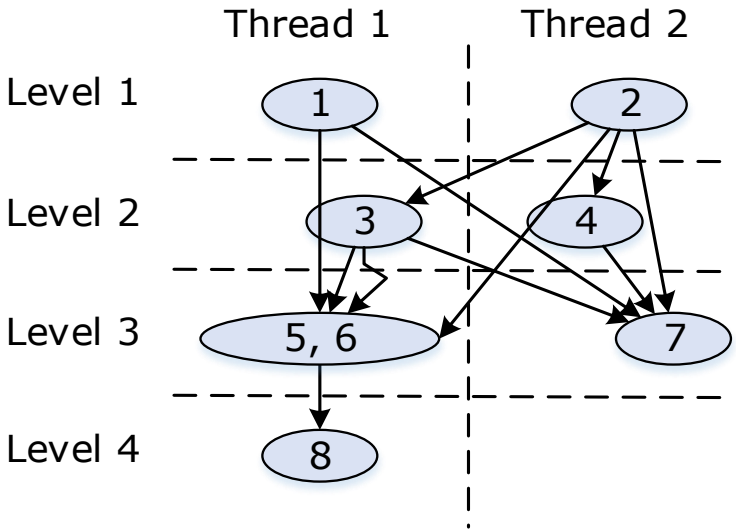
Each P2P synchronization is cheaper than barrier

But has (# of nnzs) P2P synchronizations instead of (# of levels) barrier synchronizations

- N^3 HPCG input: 27 N^3 vs. O(N)

17

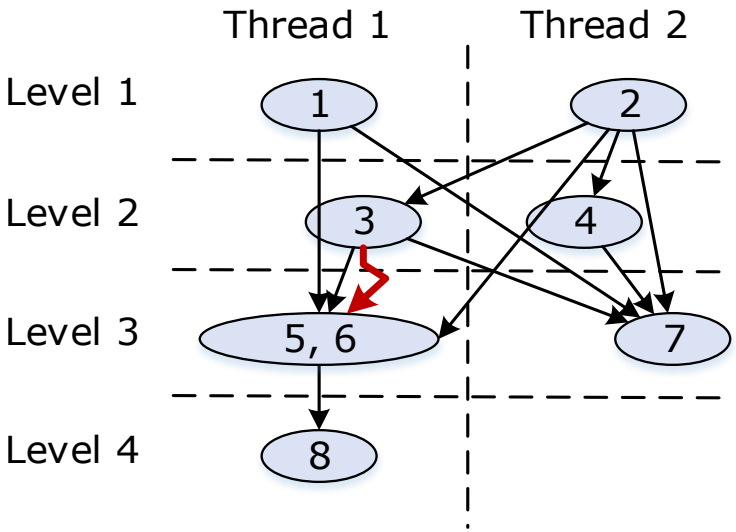
Task Coarsening



1 super-task per level per thread

18

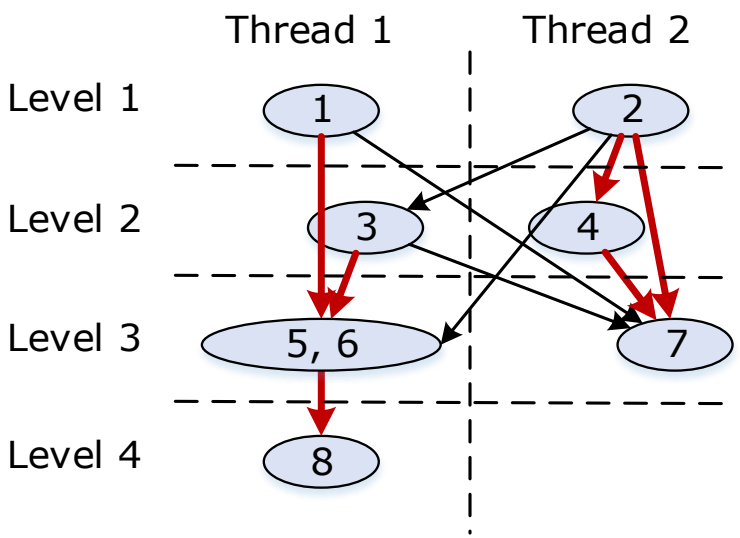
Eliminate Duplicated Edges



Duplicated edges from task coarsening

19

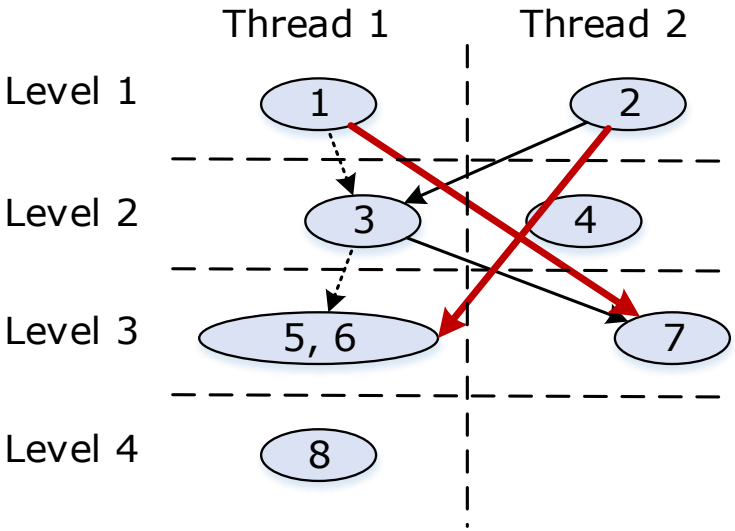
Eliminate Intra-Thread Edges



Intra-thread edges from static scheduling

20

Eliminate Transitive Edges

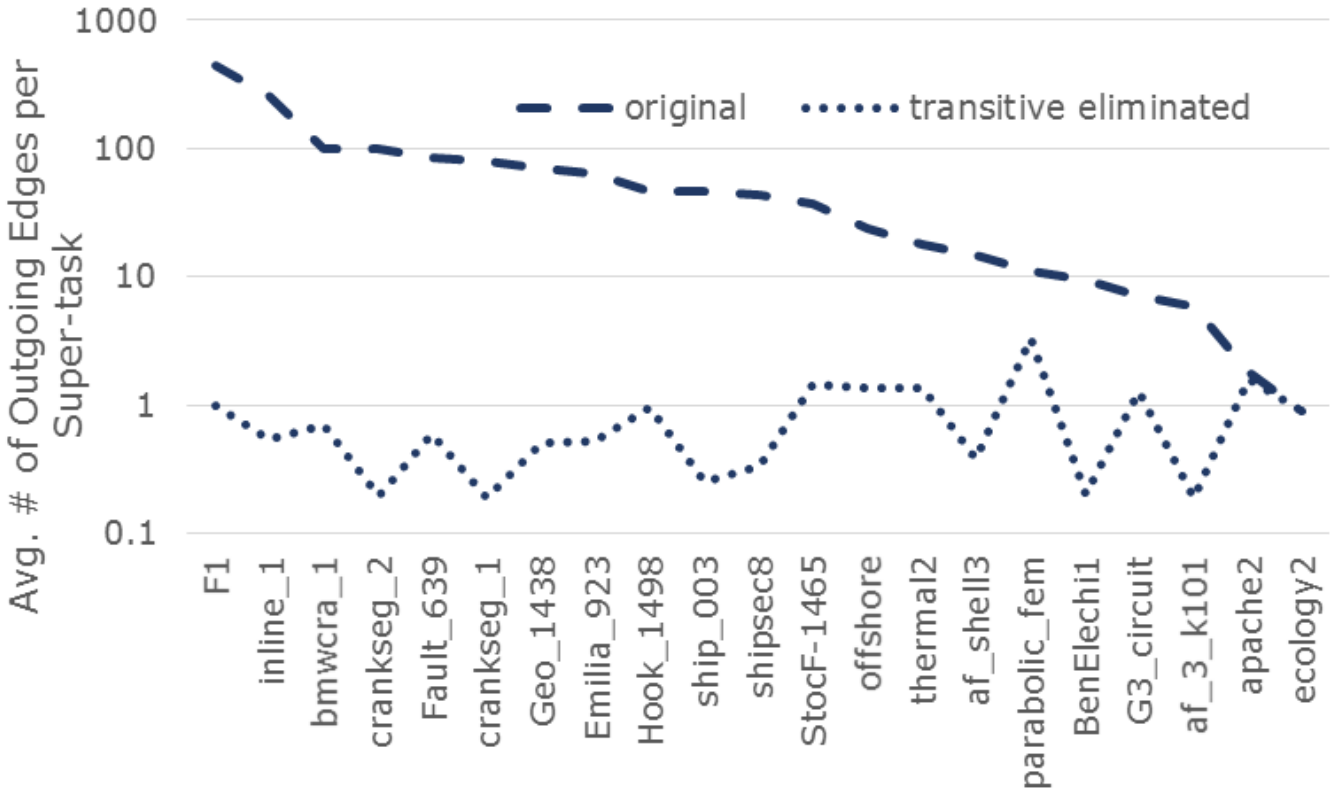


a->b, b->c, then a->c is redundant

Dotted lines: implicit schedule-imposed edges

21

Transitive edges are frequent



>90% of remaining edges (after removing duplicated and intra-thread edges) are transitive edges

22