Parallelizing GEMM within BLIS

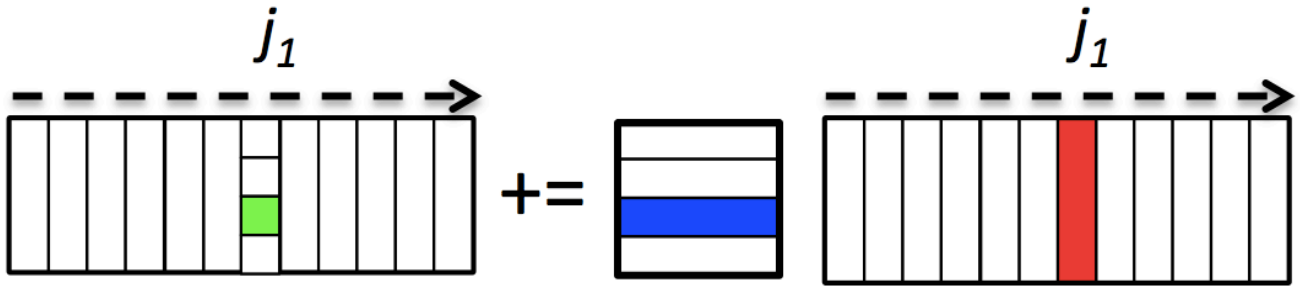
Tyler Smith

Preview

- Opportunities for parallelism within GEMM
- First attempt at parallelism
- Failures -> More sophisticated methods
- What should parallelism within BLIS look like?

Inner Kernel vs Macro-kernel

- GotoBLAS has a monolithic inner kernel

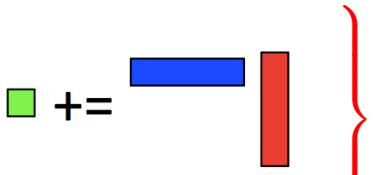


- BLIS has a macro-kernel
  - This exposes 2 additional loops
  - Can parallelize at a finer granularity

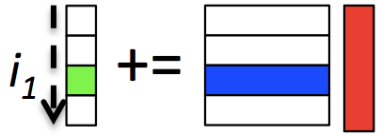
Multiple Opportunities for Parallelism

for $j_0 = 0, \ldots, n-1$ **in steps of** n_d
  for $p_0 = 0, \ldots, k-1$ **in steps of** k_c
    for $i_0 = 0, \ldots, m-1$ **in steps of** m_c
      for $j_1 = 0, \ldots, n_d-1$ **in steps of** n_r
        for $i_1 = 0, \ldots, m_c-1$ **in steps of** m_r

$$C(i_1 : i_1 + m_r - 1, j_1 : j_1 + n_r - 1) \stackrel{+}{=} \cdots$$



        endfor



(loop around micro-kernel

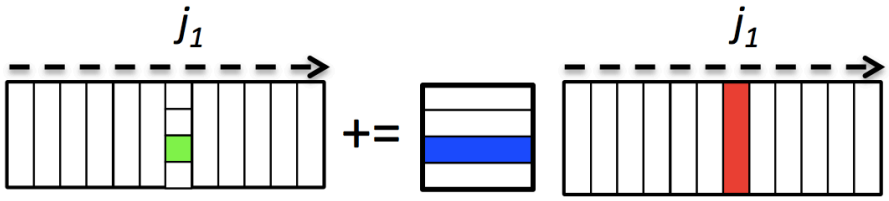
      endfor
    endfor
  endfor
endfor

Multiple Opportunities for Parallelism

for $j_0 = 0, \ldots, n-1$ **in steps of** n_d
  for $p_0 = 0, \ldots, k-1$ **in steps of** k_c
    for $i_0 = 0, \ldots, m-1$ **in steps of** m_c

      for $j_1 = 0, \ldots, n_d-1$ **in steps of** n_r
        for $i_1 = 0, \ldots, m_c-1$ **in steps of** m_r
          $C(i_1 : i_1 + m_r - 1, j_1 : j_1 + n_r - 1) \stackrel{+}{=} \cdots$
        endfor
      endfor

    endfor
  endfor
endfor

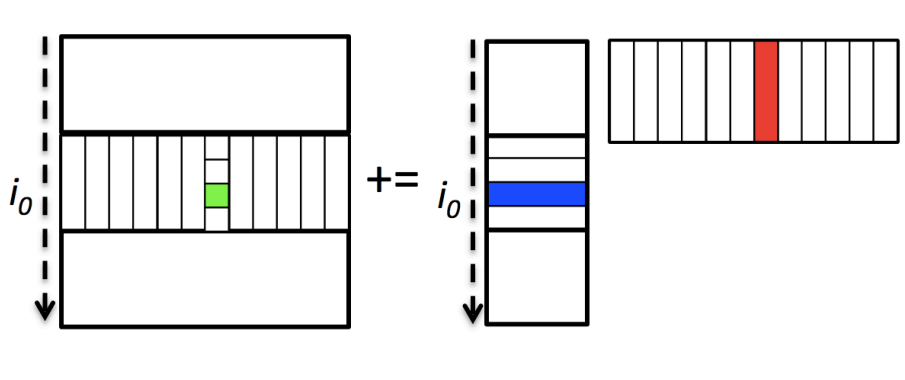


Multiple Opportunities for Parallelism

for $j_0 = 0, \ldots, n-1$ **in steps of** n_d
  for $p_0 = 0, \ldots, k-1$ **in steps of** k_c

    for $i_0 = 0, \ldots, m-1$ **in steps of** m_c
      for $j_1 = 0, \ldots, n_d-1$ **in steps of** n_r
        for $i_1 = 0, \ldots, m_c-1$ **in steps of** m_r
          $C(i_1 : i_1 + m_r - 1, j_1 : j_1 + n_r - 1) \stackrel{+}{=} \cdots$
        endfor
      endfor
    endfor

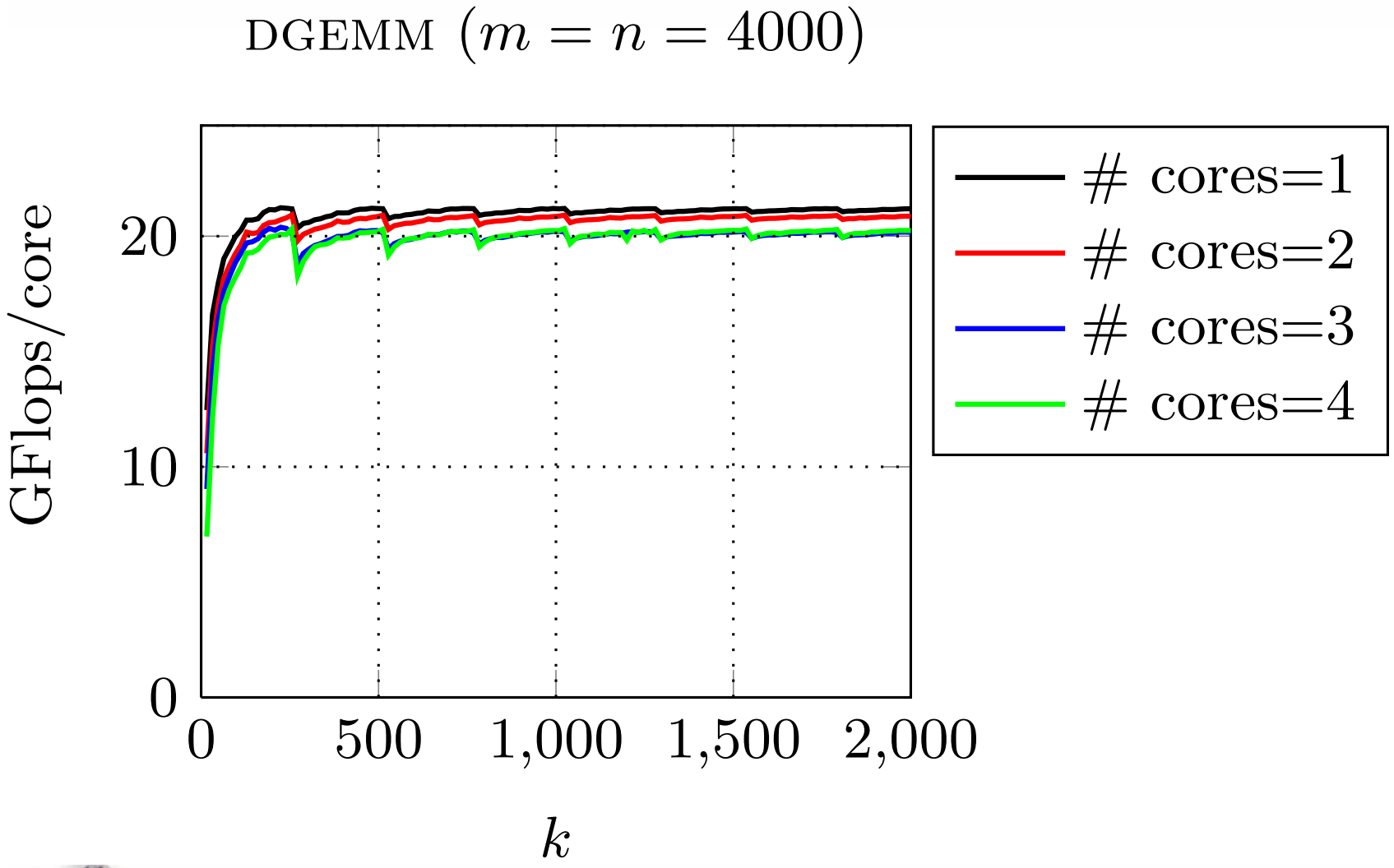
  endfor
endfor



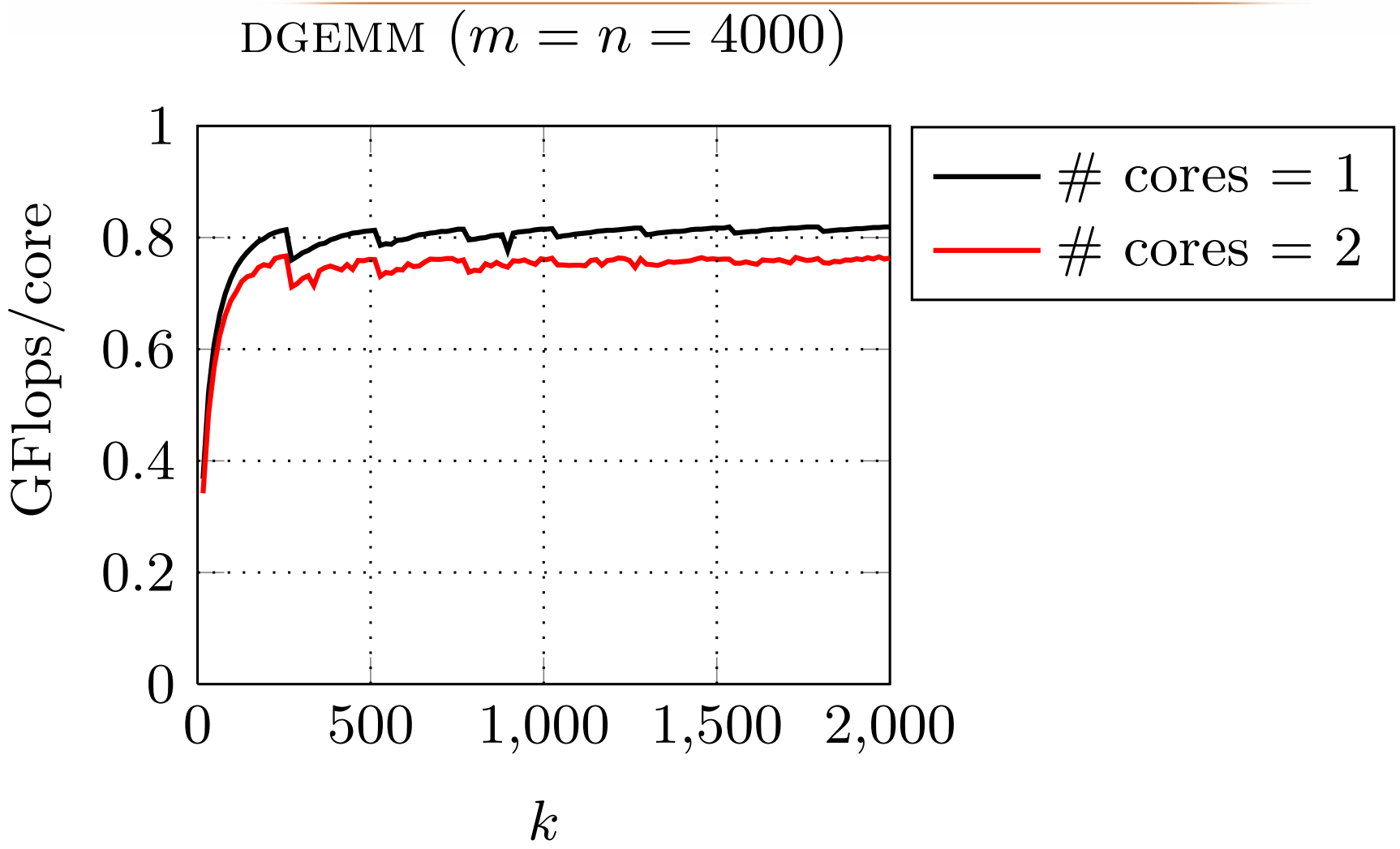
First Attempt

- Parallelized
  - 2nd loop around micro-kernel
  - Packing routines
- Simple OpenMP pragmas
- Exploit new opportunities for parallelism
- Decent performance for several architectures

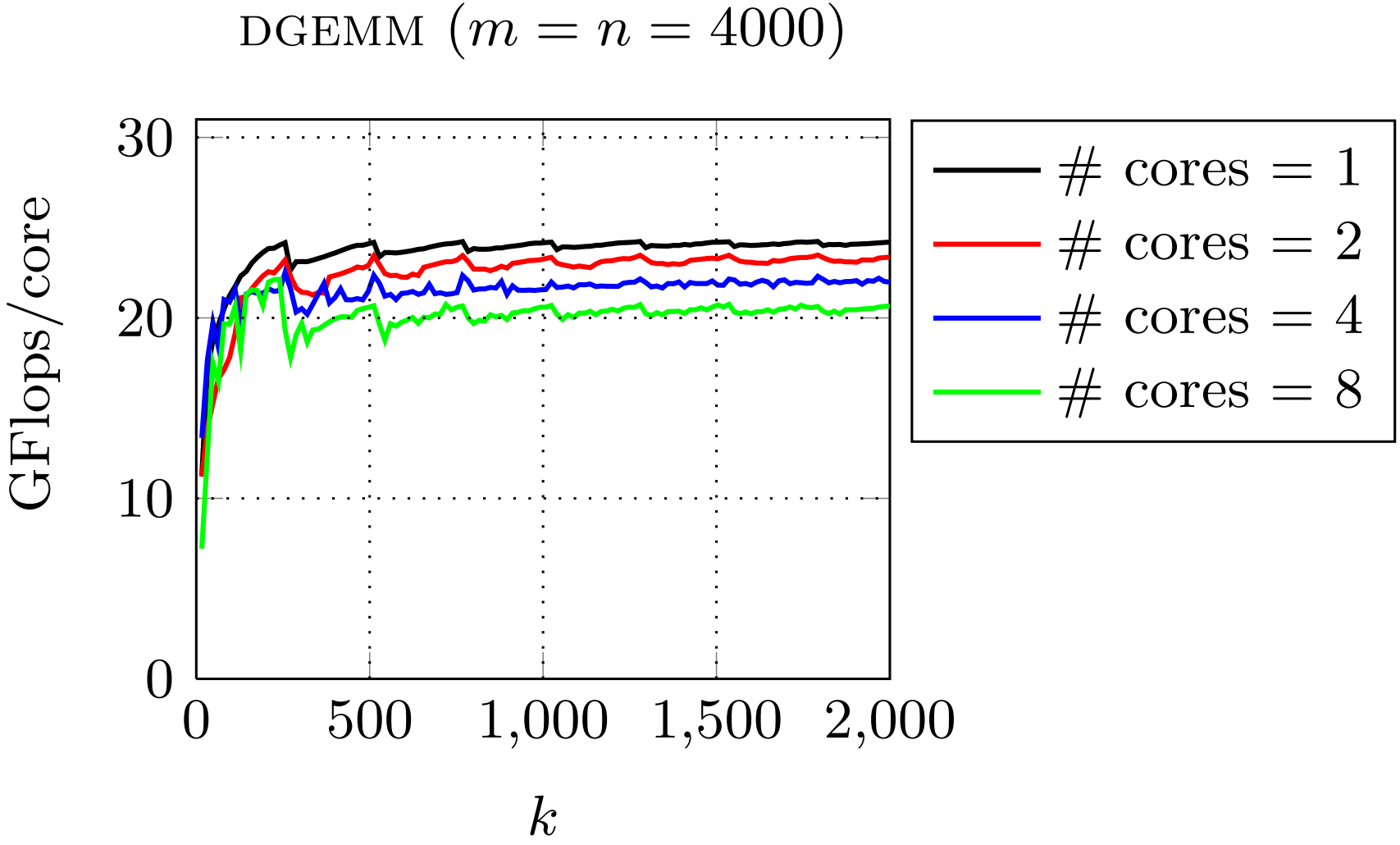
Intel Sandy Bridge E3



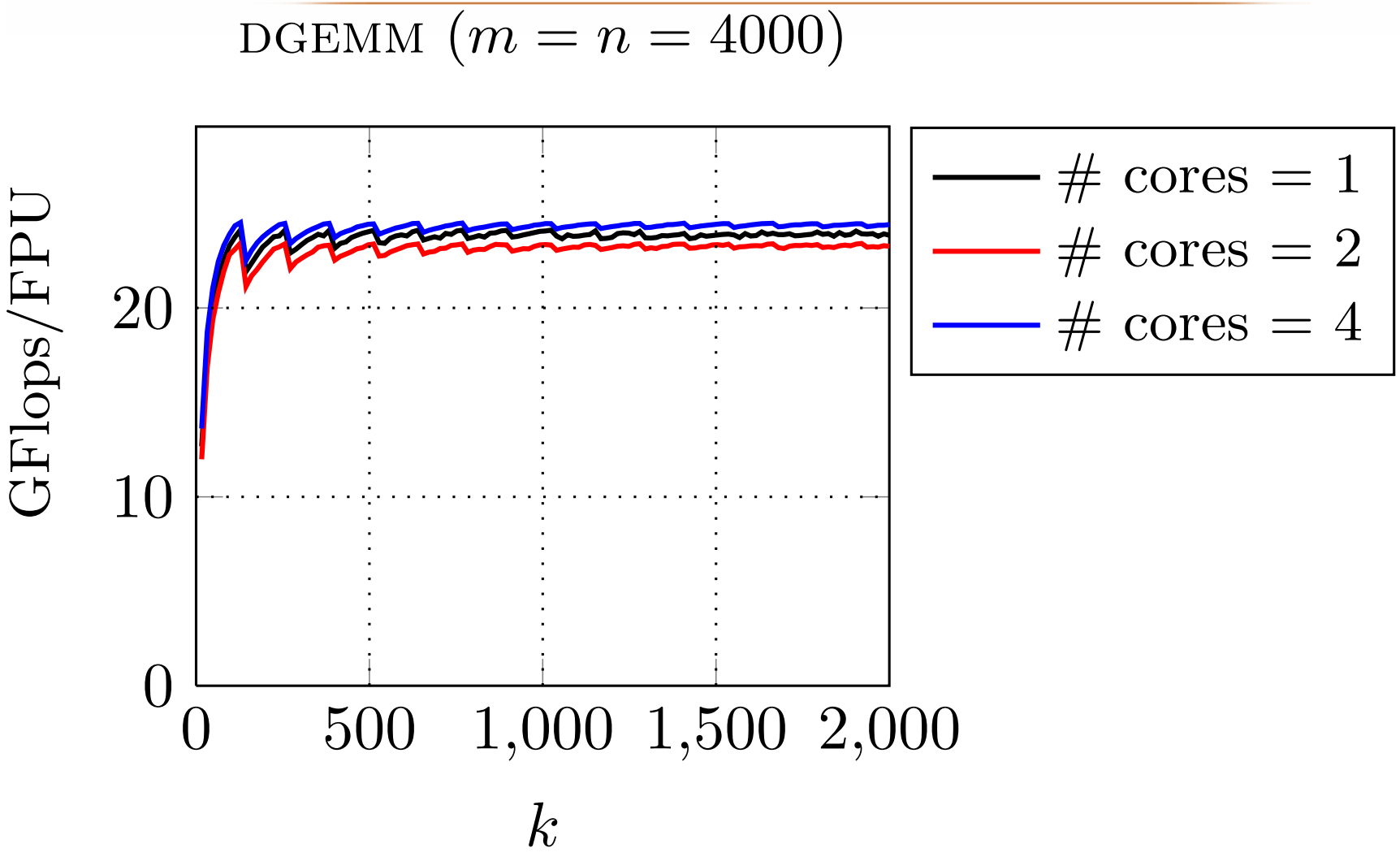
ARM Cortex A9



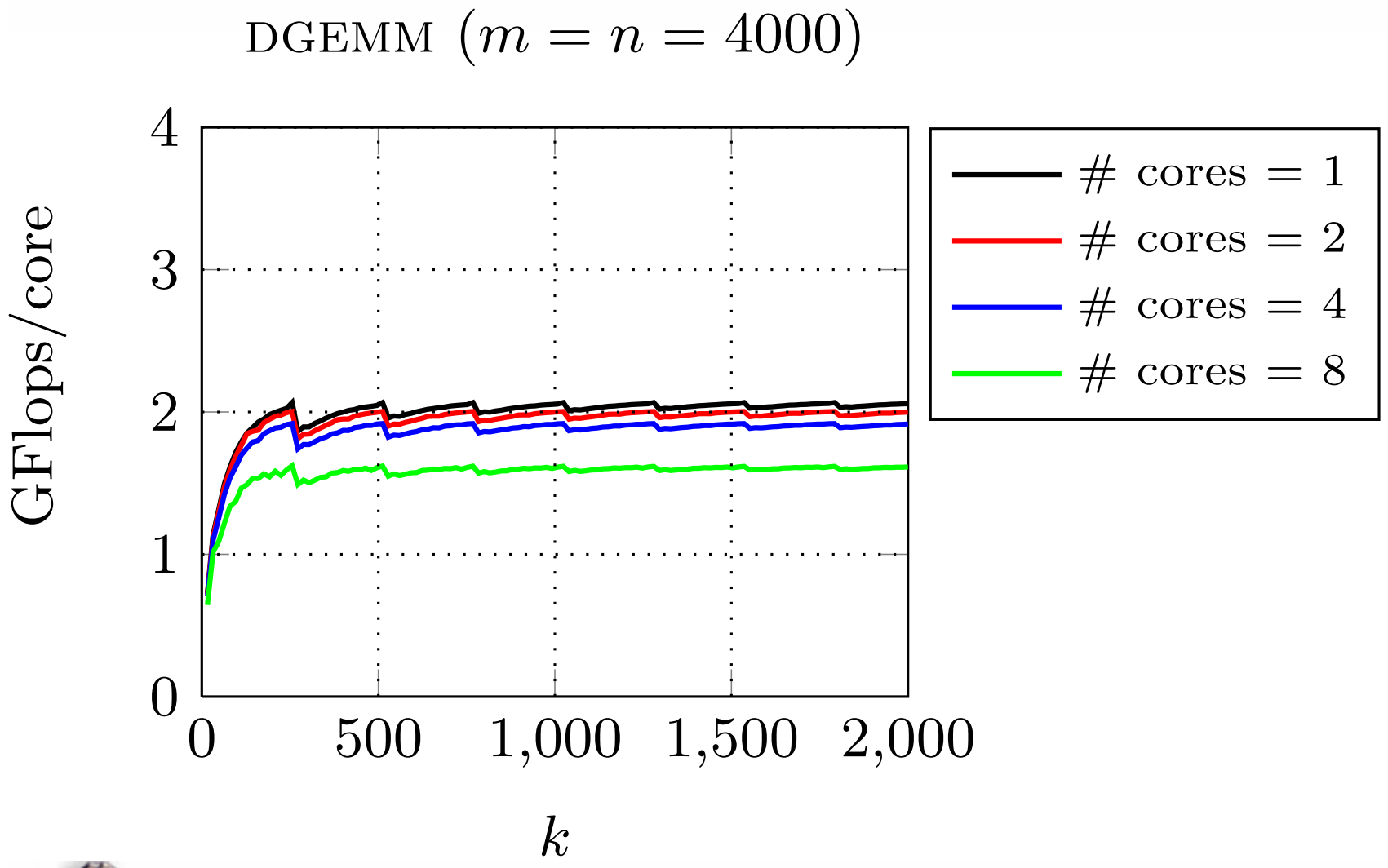
IBM Power7


DGEMM $(m = n = 4000)$
GFlops/core
30
20
10
0
0
500
1,000
1,500
2,000
k
cores = 1
cores = 2
cores = 4
cores = 8

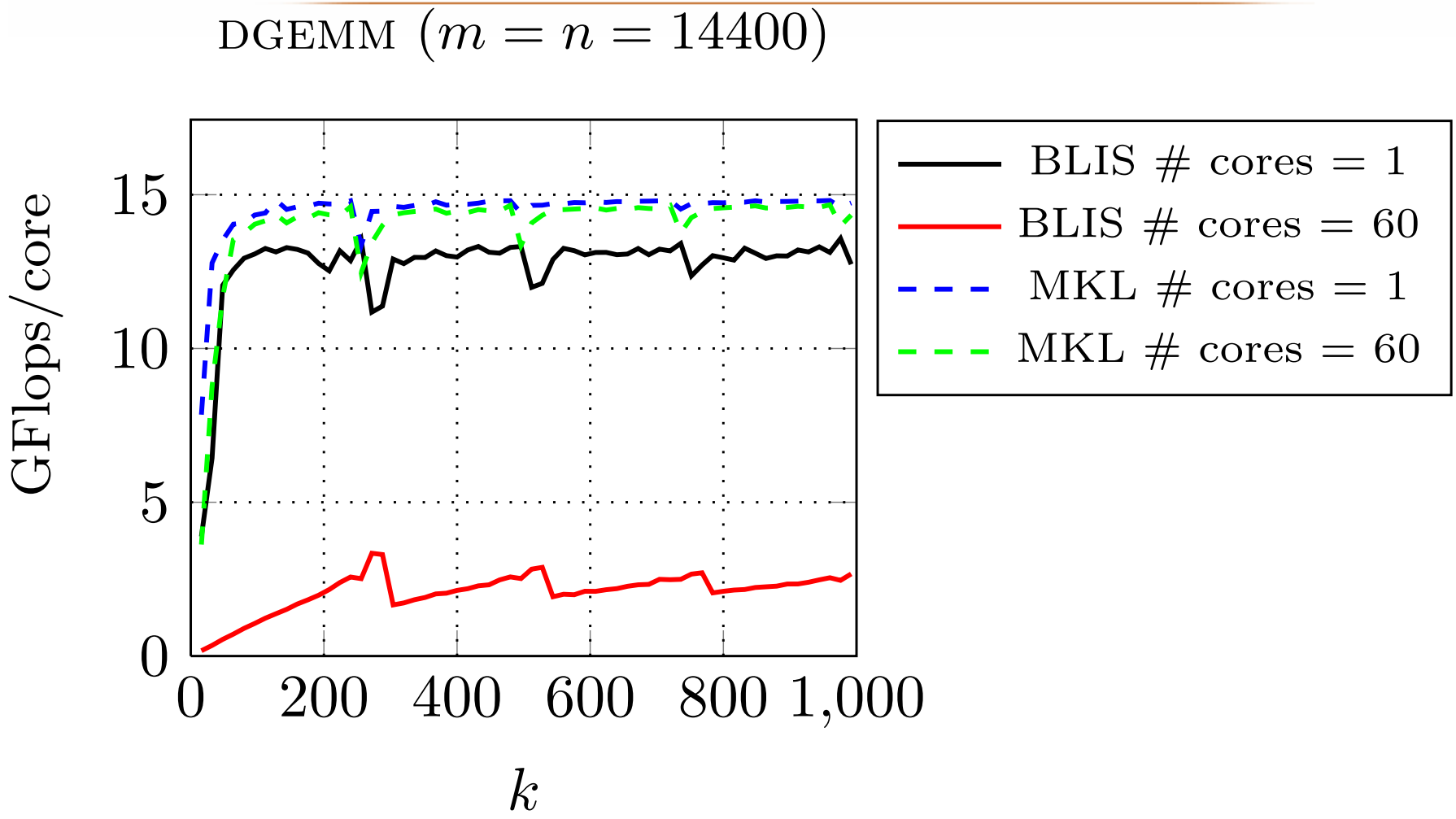
AMD A10 5800K



TI C6678 DSP


DGEMM $(m = n = 4000)$
GFlops/core
4
3
2
1
0
0
500
1,000
1,500
2,000
k
cores = 1
cores = 2
cores = 4
cores = 8

Intel Xeon Phi



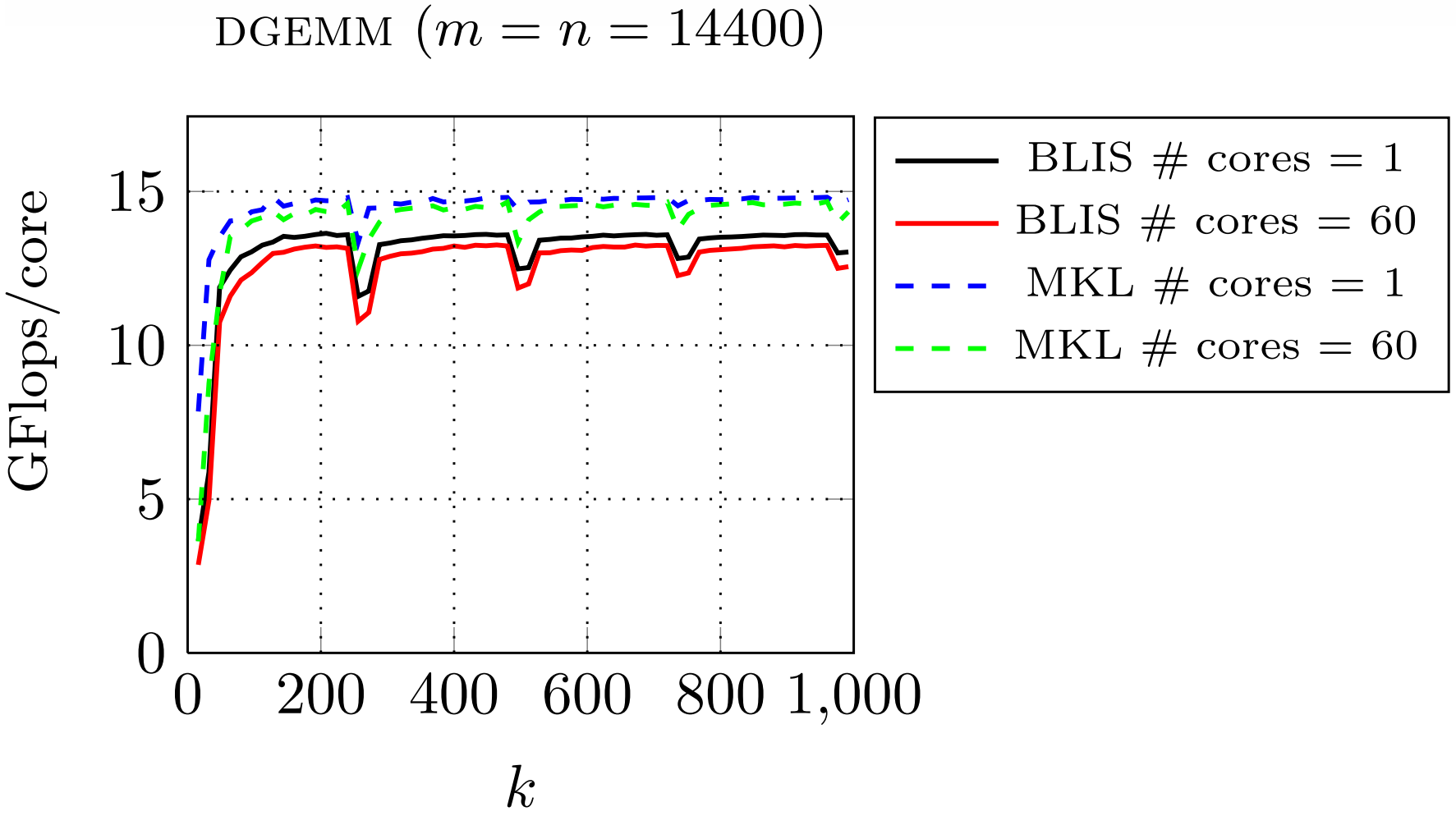
Intel Xeon Phi Problems

- Poor load balancing
  - 240 threads in the m dimension
- Poor amortization
  - Each block of A is moved into the L2 cache, only used with 7 or 8 slivers of B
- Poor use of L1 cache

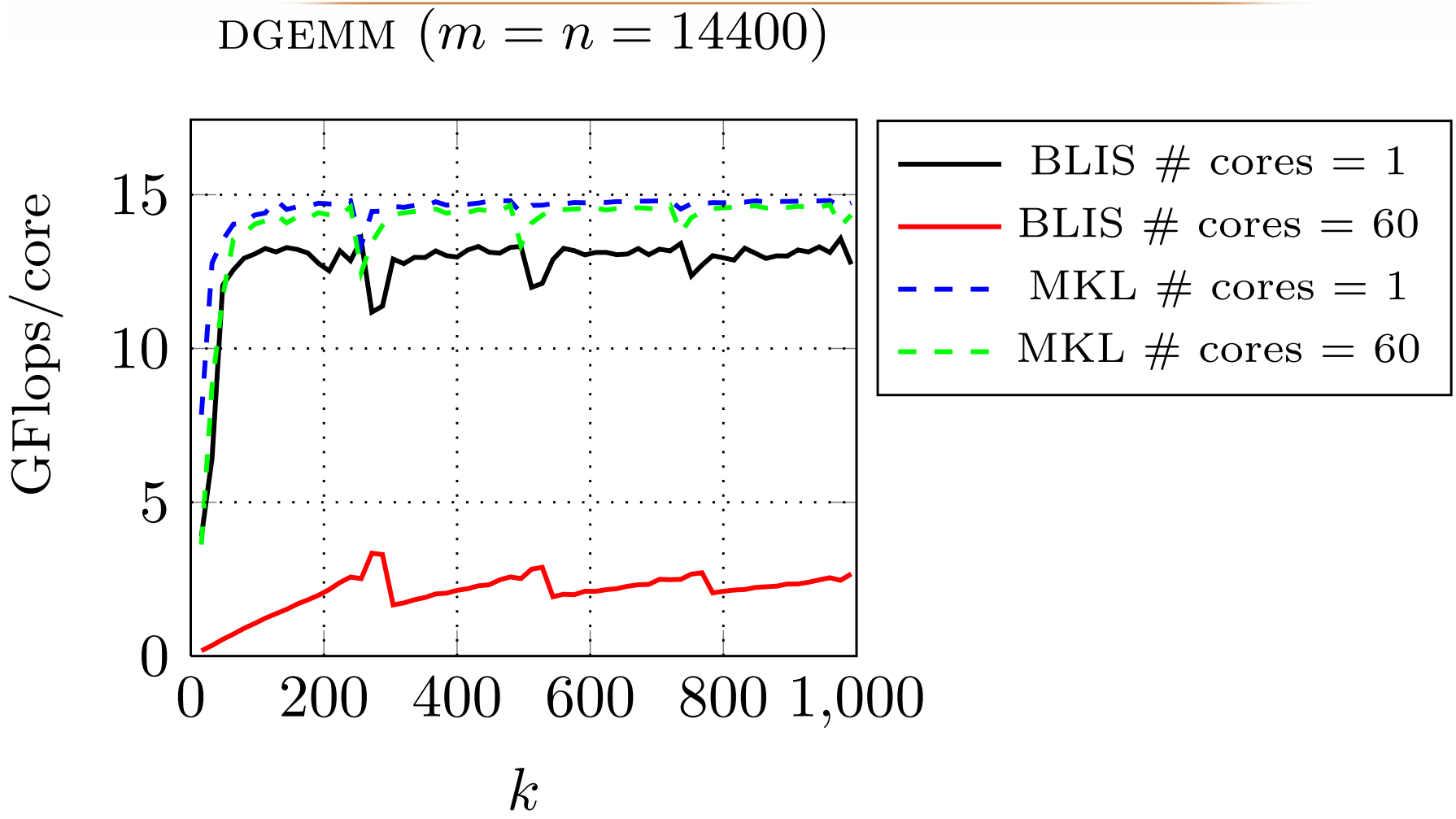
Intel Xeon Phi

- Parallelize 3rd loop around microkernel
  - Split the loop between 60 cores
- Parallelize 2nd loop around microkernel
  - Split the loop between 4 threads
- Synchronize hardware threads

Intel Xeon Phi



Intel Xeon Phi

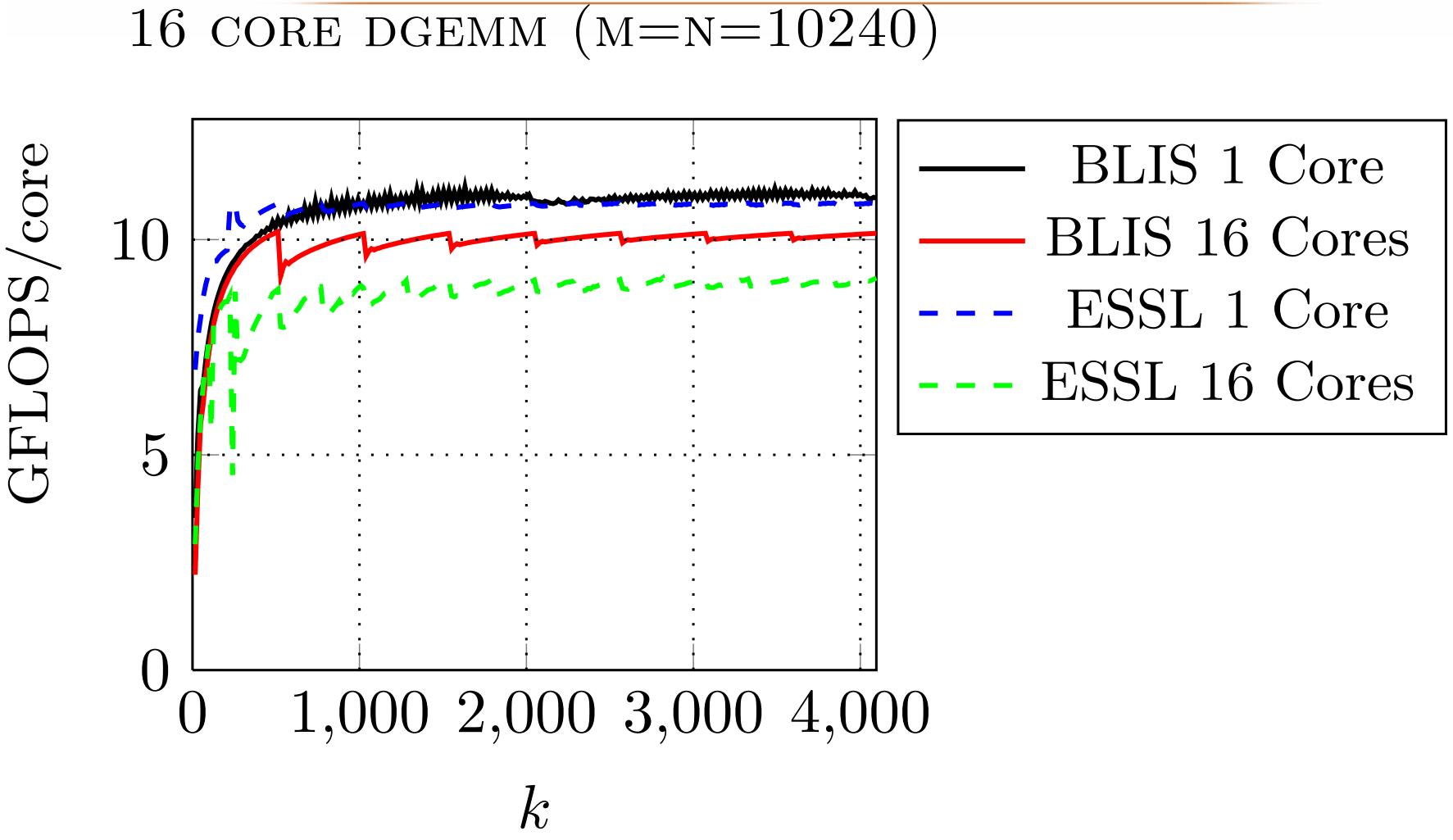


What if we do the same for BG/Q

- Very similar to Xeon Phi
  - Low Power
  - Many Core
  - In-order
  - Multiple threads per core

	BG/Q	MIC
# Cores	16	60
Threads per Core	4	4
# L2 Caches	1	60
L2 Cache Size	32 MB	512 KB

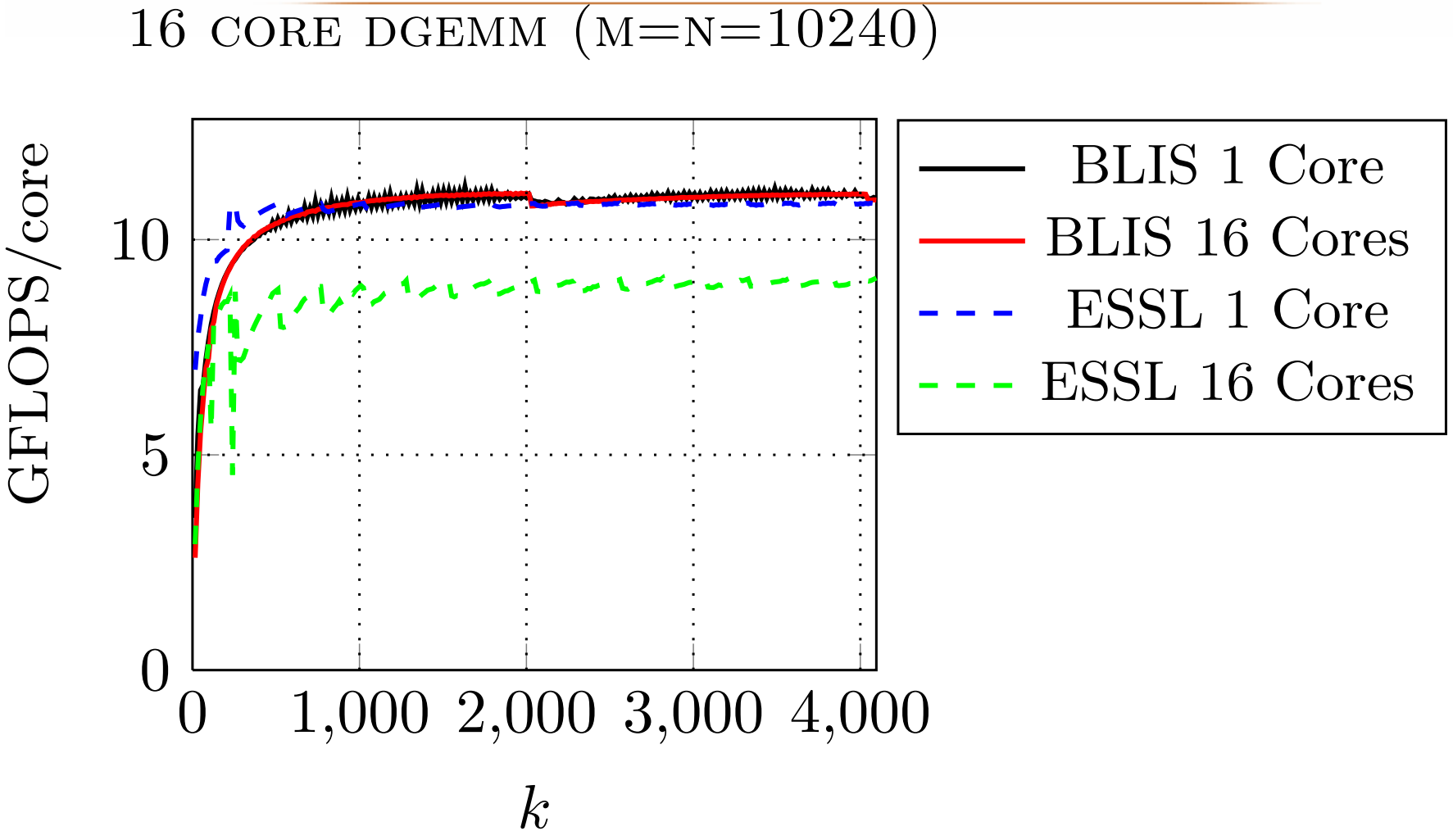
IBM Blue Gene/Q



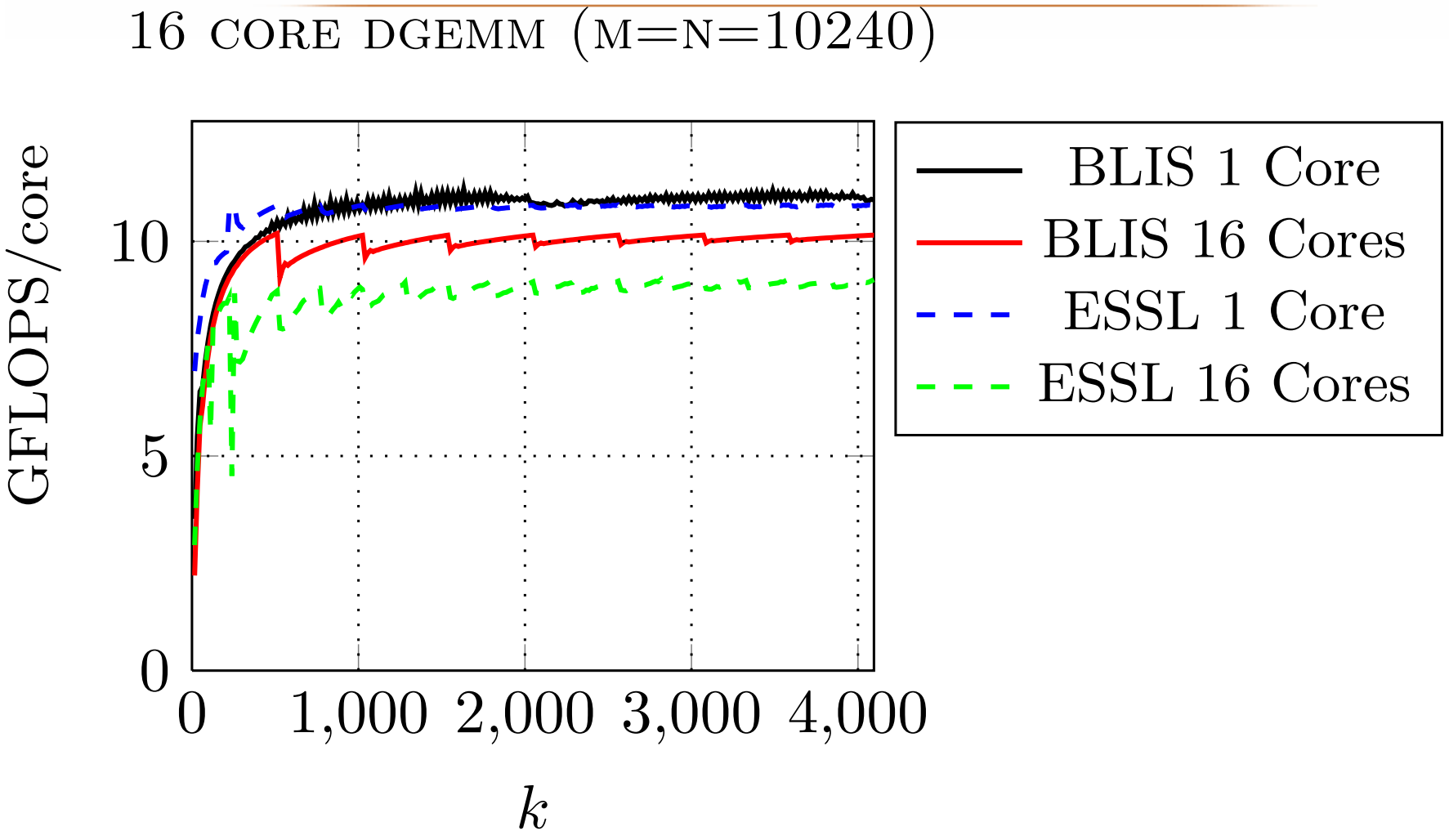
Blue Gene/Q

- Large, shared L2 cache
  - Problem:
    - Parallelizing 3rd loop means it holds multiple blocks of A
    - Decrease Size of A
    - Lower flops to memops ratio
  - Solution
    - Parallelize 1st and 2nd loops around micro-kernel

IBM Blue Gene/Q



IBM Blue Gene/Q



Requirements for Multithreaded BLIS

- Multiple levels of parallelism
- Map level of parallelism to hardware
  - We have some hypothesis on how to do this
  - Work in progress

Experimental Implementation

- Multiple levels of parallelism
- Hierarchical Groups
  - Conceptually similar to nested OpenMP parallelism
  - Use thread communicators to share data within groups

Experimental Implementation

- Thread Communicators
- Inspired by MPI Communicators
  - Provide:
    - Barriers
    - Locks
    - Broadcast

Future Work

- Extend to other BLIS operations
  - Bryan's work within DxT
- High level interface
  - Express cache hierarchy
  - Direct BLIS on what hardware to use
- Analyze what levels to parallelize and why