Porting optimized HPCG 3.1 for IBM BG/Q to IBM POWER9

Optimization overview and main results

Panagiotis Chatzidoukas, Cristiano Malossi, Christoph Hagleitner, Costas Bekas
IBM Research – Zurich

November 13th, 2018

IBM

Analysis of HPCG 2.4 on BG/Q: no optimization

- System: Juqueen IBM BG/Q, 28 racks, 458752 cores
- HPCG local domain dimension: 80x80x80
- Result: 59.1 TFlop/s (2 GFlops/s per node)

Kernel	Time [%]	GFlop/s
DOT	1.5	62997.8
WAXPY	1.34	69182.5
SPMV	14.4	59170
MG	82.78	57495.1
Raw total	100	57971.4
Total	-	59154.4

Key observations:

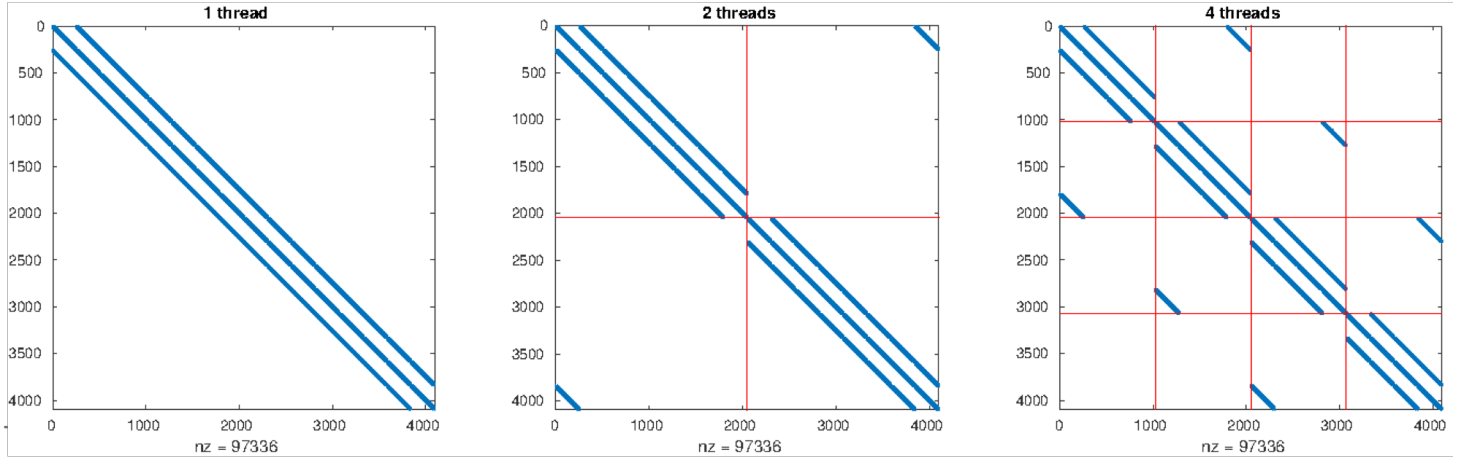
- 82 % of time is MG (no threads in ref. implementation)
- MG main routine need to load 15 to 20 Bytes for each 2 Flops: Bytes/Flop = 7.5 to 10
- BG/Q has 30 GB/s bandwidth to memory, so we expect 3 to 4 GFlop/node

Optimizations (1) – Smart pivoting for parallel SYMGS

- BG/Q has 4 threads per core
- A parallel implementation of SYMGS requires coloring
- Coloring has two undesired side effects: 1) slow down convergence, 2) limit cache reuse

Our solution:

- Stencil discretization lead to a uniform (diagonal) matrix structure: we can rely on that!



Optimizations (2) – Fine tuning

- Strong compiler optimization (-O5, -qipa=level=2, -qhot=level=2, etc., little effect overall)
- AXPY and DOT manually SIMD vectorized (slightly better performances than auto-compiled versions)
- Contiguous storage for matrix (to help hardware prefetching – very important on BG/Q)
- Improve backward prefetching of Gauss-Seidel smoother with __dcbt instructions
- Manual unrolling factor of 2 for SpMV, and slightly improved code
- Use Isend/Irecv with a single MPI_Waitall call at the end (better overlap of communications)

and, not less important, optimized local problem size

- A smaller problem size gives better MG and SPMV performance (better cache use)
  ... however it also increases DOT MPI_Allreduce (due to wait time) – need to find the best compromise!
- We use 48x16x16 or 56x16x16 (on few racks) and 96x32x32 or 112x32x32 (on many racks)

HPCG 2.4 on BG/Q benchmark (reference vs optimized)

Non-optimized:

- 32 MPI task/node, no threads
- Local domain dimension: 80x80x80
- 59.1 Tflops/s (2 GFlop/s per node)

Kernel	Time [%]	GFlop/s
DOT	1.5	62997.8
WAXPY	1.34	69182.5
SPMV	14.4	59170
MG	82.78	57495.1
Raw total	100	57971.4
Total	-	59154.4

Optimized:

- 16 MPI task/node, 4 threads each
- Local domain dimension: 112x32x32
- 95.5 Tflop/s (3.33 GFlop/s per node)

Kernel	Time [%]	GFlop/s
DOT	3.68	42098.9
WAXPY	1.5	103352
SPMV	14.11	100137
MG	80.7	98033.7
Raw total	100	96333.2
Total	-	95476.4

HPCG 3.1 on Vulcan & Sequoia

Vulcan (24K nodes):

- 32 MPI tasks/node, 2 threads each
- Local domain dimension: 112x32x32
- 80.9 TFlop/s (3.29 GFlop/s per node)

Kernel	Time [%]	GFlop/s
DOT	6.5	21420.3
WAXPY	2.0	69445.9
SPMV	14.2	88960.5
MG	77.3	91263.6
Raw total	100	85980
Total	-	80893.7

Fraction of peak = 1.6%

Sequoia (96K nodes):

- 32 MPI task/node, 2 threads each
- Local domain dimension: 112x32x32
- 330.4 Tflop/s (3.36 GFlop/s per node)
  - #10 in HPGC results list (June 2018)

Kernel	Time [%]	GFlop/s
DOT	3.8	149613
WAXPY	1.8	324545
SPMV	15.3	343314
MG	79.1	370952
Raw total	100	357356
Total	-	330373

Fraction of peak = 1.6%

perfect scaling from 24K to 96K nodes

Changes in optimizations for POWER9

- The XLC (16.1.0) compiler optimizations for BGQ remain the same - still little effect overall
- MPI configuration
  - One task per core
  - Binding policy: *mpirun --bind–to core --map-by core*
- OpenMP configuration:
  - two threads per task (core)
  - *OMP_PROC_BIND=FALSE* (no explicit binding to the hardware threads of the core)
  - *OMP_WAIT_POLICY=ACTIVE* (no need for yield)
- Problem size
  - Local domain is set to 160x160x96 (or 160x96x160)
  - Larger than BGQ due to higher memory bandwidth

MareNostrum P9 CTE @ BSC

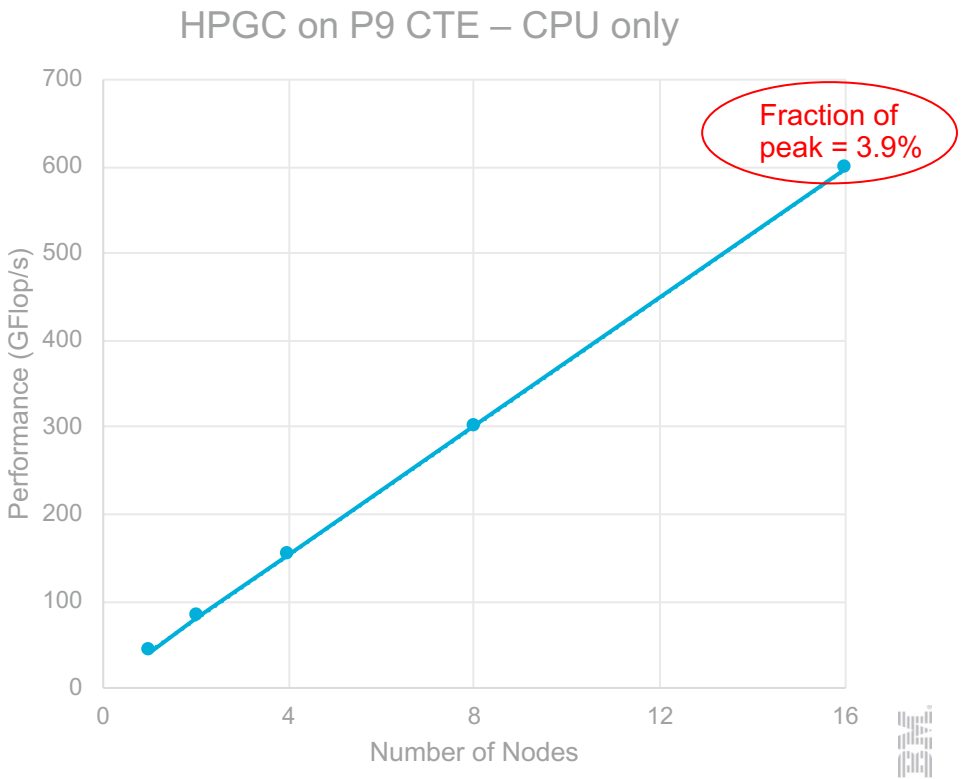
- #255 in TOP500 (June 2018)
  - https://www.top500.org/system/179442
- 54 nodes organized in 3 racks
  - 52 compute and 2 login nodes
- Each node
  - 2x IBM POWER9 20C 3.1GHz
    - 40 cores with four-way multithreading variant SMT4
  - 4x NVIDIA Tesla V100
- Interconnection network: Dual-rail Mellanox EDR Infiniband

HPCG 3.1 on MareNostrum POWER9 CTE

MareNostrum IBM POWER9 CTE (16 nodes):

- 40 MPI tasks/node, 2 threads each
- Local domain dimension: 160x160x96
- 597 GFlop/s (37.3 GFlop/s per node)

Kernel	Time [%]	GFlop/s
DOT	5.8	173.3
WAXPY	2.7	374.4
SPMV	14.4	636.3
MG	77.1	666.3
Raw total	100	625.4
Total	-	597.4

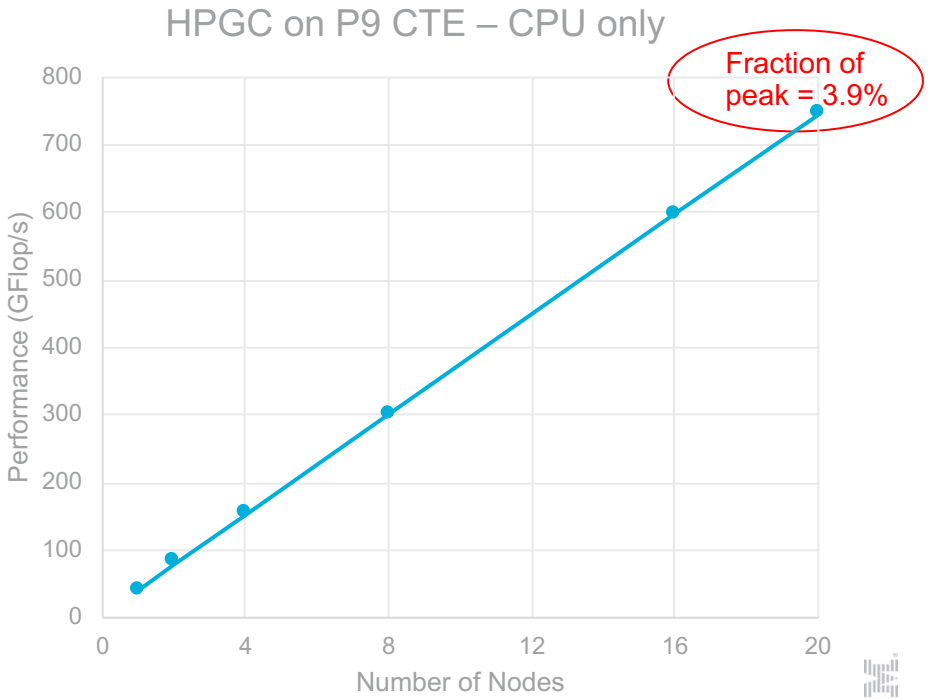


HPCG 3.1 on MareNostrum POWER9 CTE

MareNostrum IBM POWER9 CTE (20 nodes):

- 40 MPI tasks/node, 2 threads each
- Local domain dimension: 160x160x96
- 744.6 GFlop/s (37.2 GFlop/s per node)

Kernel	Time [%]	GFlop/s
DOT	3.5	359.9
WAXPY	2.7	475.4
SPMV	14.9	771.0
MG	78.9	815.7
Raw total	100	783.9
Total	-	744.6

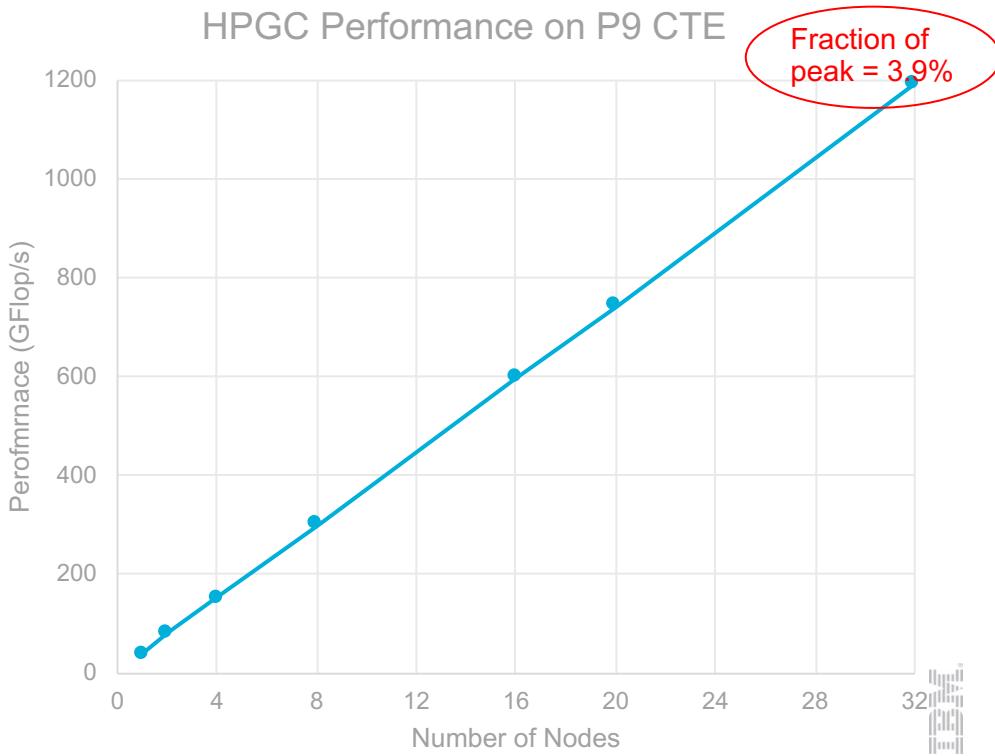


HPCG 3.1 on MareNostrum POWER9 CTE

MareNostrum IBM POWER9 CTE (32 nodes):

- 40 MPI tasks/node, 2 threads each
- Local domain dimension: 160x160x96
- 1193.4 GFlop/s (37.3 GFlop/s per node)

Kernel	Time [%]	GFlop/s
DOT	6.5	311.8
WAXPY	2.7	740.4
SPMV	14.5	1268.7
MG	76.4	1342.8
Raw total	100	1249.0
Total	-	1193.4



Conclusions

- An optimized CPU-only version of HPCG on IBM processors
  - explicit SIMD vectorization, data prefetching, asynchronous MPI communication
  - smart pivoting: new OpenMP parallelization approach for SYMGS, the most time consuming kernel of HPCG
- Fine tuning of several parameters, such as
  - Local problem size
  - Number of MPI tasks and OpenMP threads
- The code achieves 1.6% of the peak performance on IBM BGQ and 3.9% on IBM POWER9
- The code will be released soon as open source project

THANK YOU