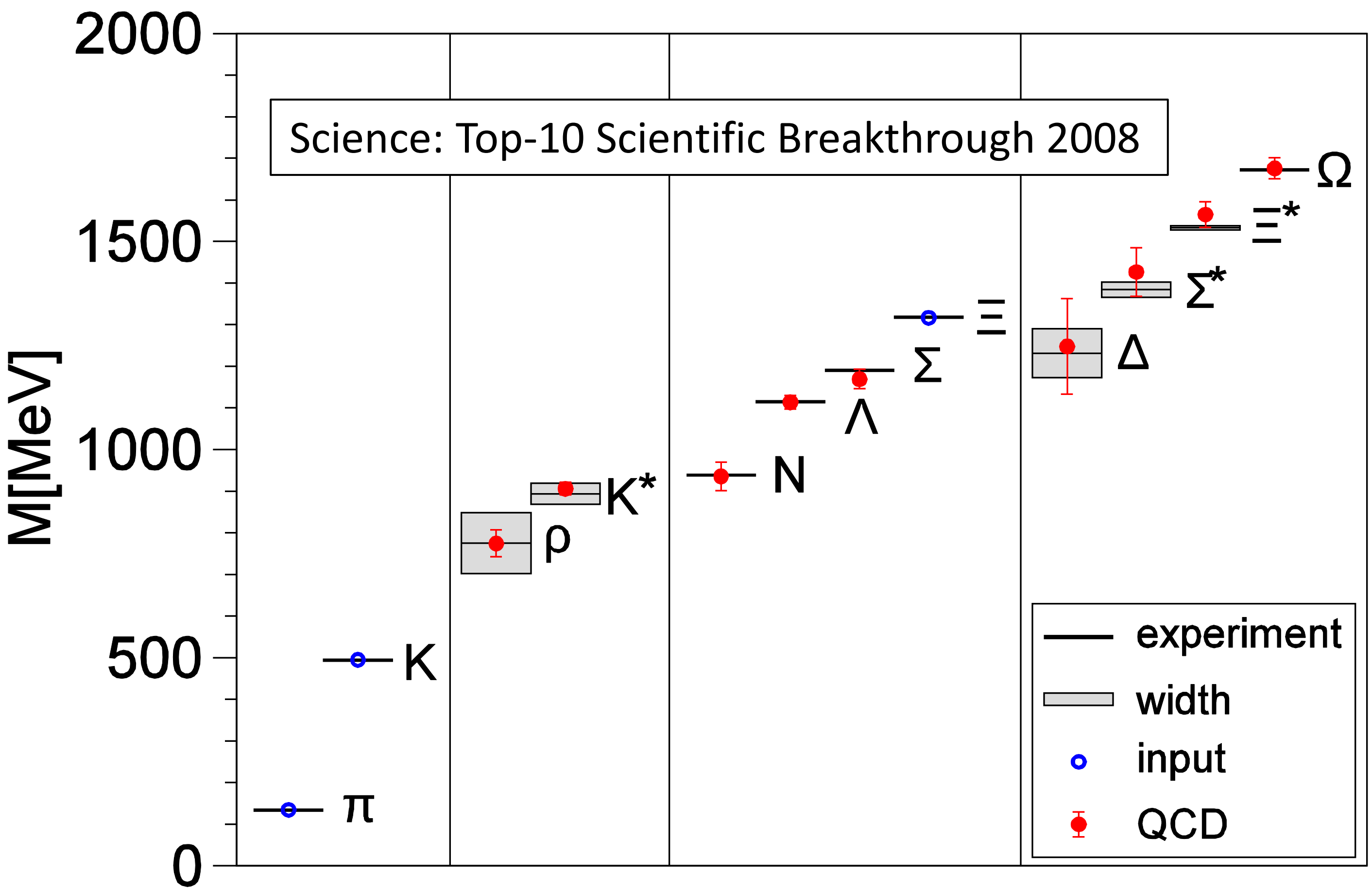
Experiences with Lattice QCD on the Blue Genes at Juelich

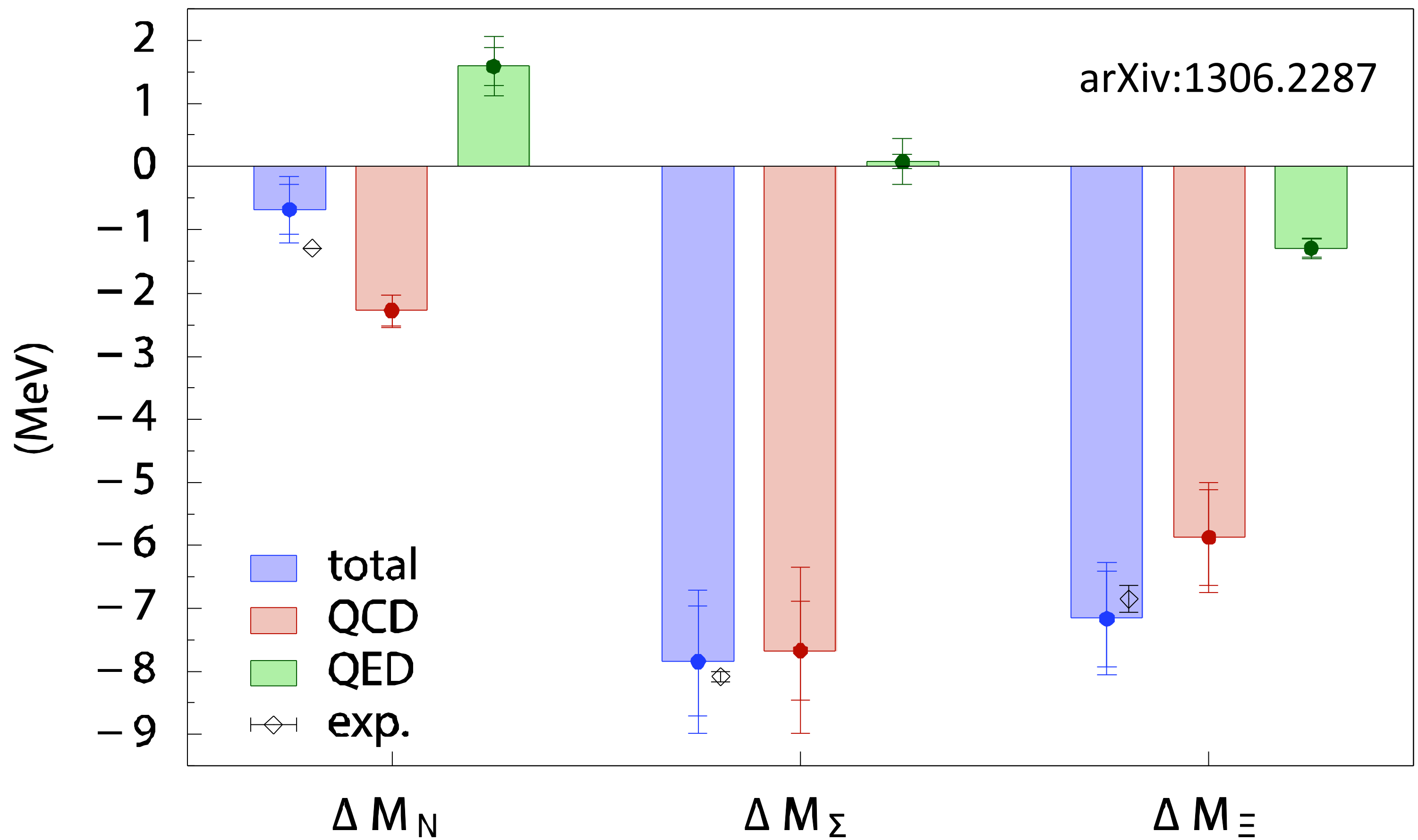
10. October 2013 | Stefan Krieg

Motivation: Physics results from LQCD on BG

Budapest-Marseille-Wuppertal Coll., Science **322**, 1224



Motivation: Physics results from LQCD on BG



The new machine

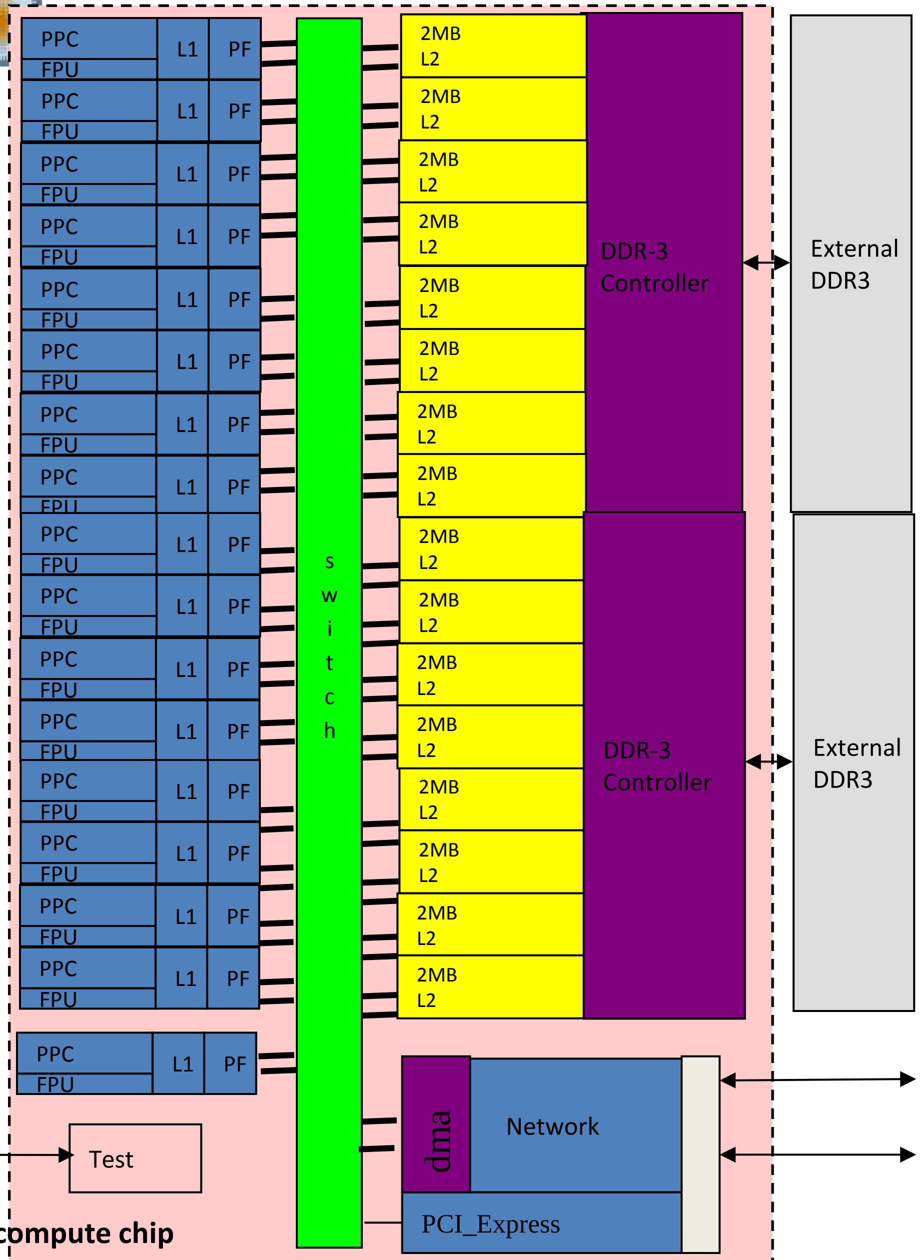
Specs:

- 28 Racks
- 7x2x2Racks
  = **28**x8x8x8x2 nodes
  = 28,672 nodes
  = 458,752 cores
  = 1,835,008 HW tds.
- 5.9 Pflop
- Top500 #7
- Public: PRACE, GCS, NIC local use (JARA)
- Production: Jan. 2013 (8->16->24->28)

BERGISCHE UNIVERSITÄT WUPPERTAL

JÜLICH FORSCHUNGSZENTRUM

BGQ Chip architecture



- 16+1 core SMP
  - Each core 4 way hardware threaded
- Transactional memory and thread level speculation
- Quad float point unit on each core
  - 204.8 GF peak node
- Frequency target of (1.6) GHz
- 563 GB/s bisection bandwidth to shared L2
  - (BGL at LLNL has 700 GB/s system bisection)
- 32 MB shared L2 cache
- 42.6 GB/s DDR3 bandwidth
  - (2 channels each with chip kill protection)
- 10 intrarack interprocessor links each at 2.0GB/s
- 1 I/O link at 2.0 GB/s
- 4-8 GB memory/node
- ~30 Watts chip power

2 GB/s I/O link (to I/O subsystem)

10*2GB/s Intrarack (5-D torus)

** chip I/O shares function with PCI_Express

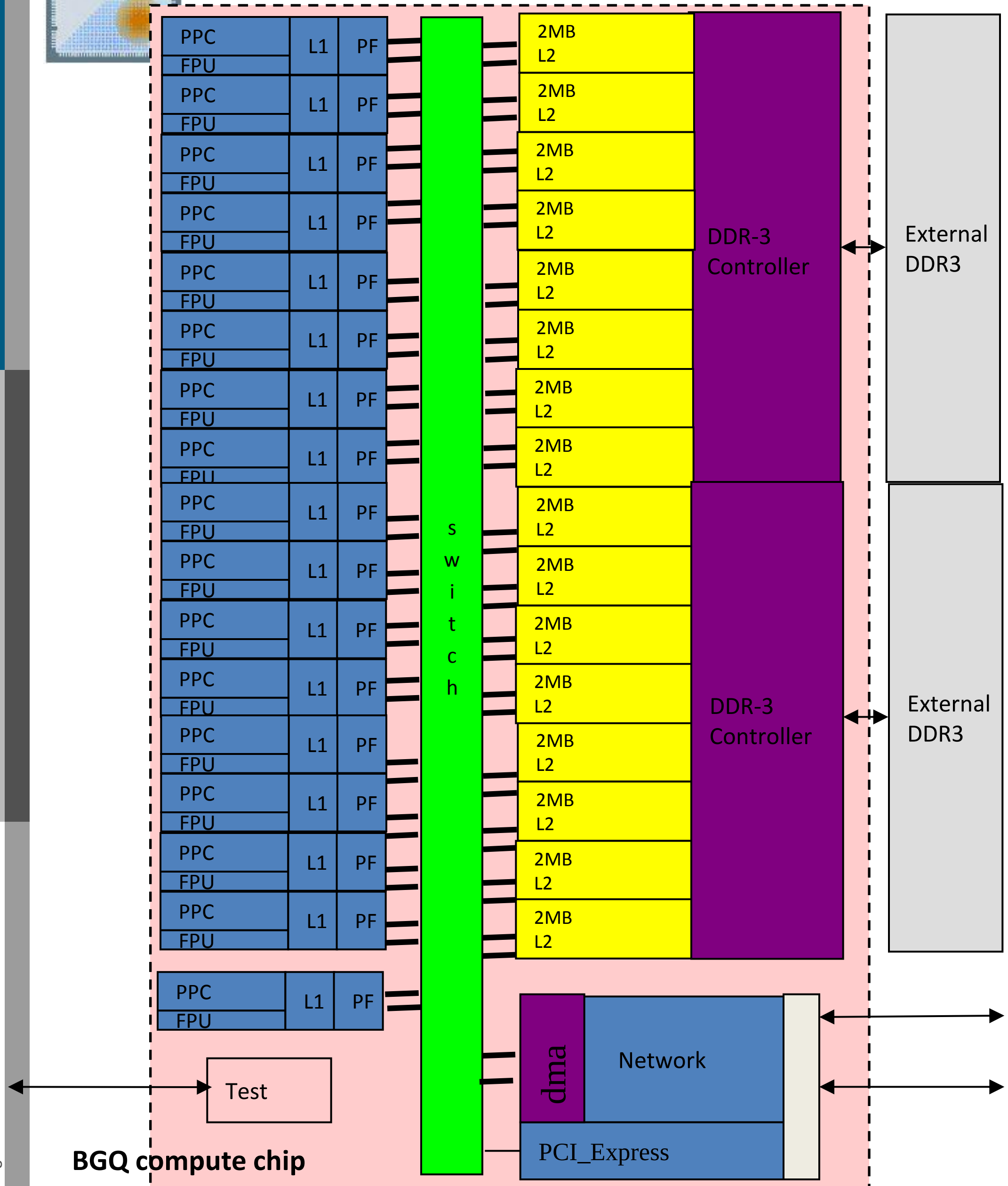
Blue Gene/Q vs. Blue Gene/P

Blue Gene/Q	Blue Gene/P
• Midplane = 105 Tflop/s • 32768 HW threads • = 3.2 Gflop/s per thread • 5d interconnect: • 512 = 2x4x4x4x4	• 16 MP = 111 Tflop/s • 32768 HW threads • = 3.4 Gflop/s per thread • 3d interconnect: • 8192 = 16x16x32

→ **USE master threads if possible!**

Comparison of relevant hardware specs

Blue Gene/P	Blue Gene/Q	Q vs. P
300.000 threads	6.291.456 threads	21
1 Pflop/s	20 Pflop/s	20
Cache: 2.0 MB/core	Cache: 2.0 MB/core	1
Cache: 109 GB/s BW	Cache: 563 GB/s BW	6
Mem: 0.5 GB/thread	Mem: 0.5 GB/thread	1
Mem: 13.6 GB/s BW	Mem: 42.6 GB/s BW	3
FPU: 4 Flop/c/core	FPU: 8 Flop/c/core	2
FPU: 13.6 Gflop/s/node	FPU: 204.8 Gflop/s/node	15



BGQ Chip architecture

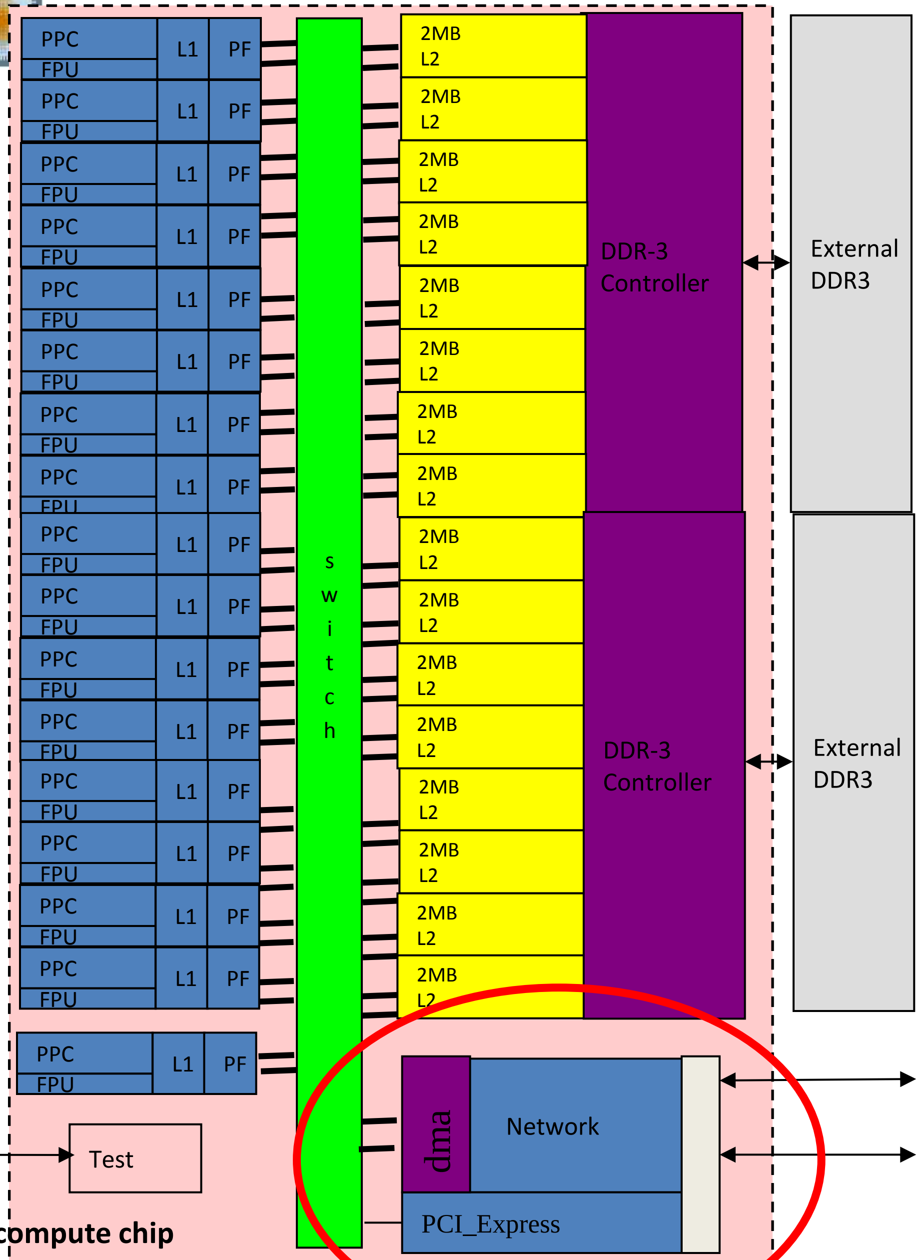
- 16+1 core SMP
  - Each core 4 way hardware threaded
- Transactional memory and thread level speculation
- Quad float point unit on each core
  - 204.8 GF peak node
- Frequency target of (1.6) GHz
- 563 GB/s bisection bandwidth to shared L2
  - (BGL at LLNL has 700 GB/s system bisection)
- 32 MB shared L2 cache
- 42.6 GB/s DDR3 bandwidth
  - (2 channels each with chip kill protection)
- 10 intrarack interprocessor links each at 2.0GB/s
- 1 I/O link at 2.0 GB/s
- 4-8 GB memory/node
- ~30 Watts chip power

2 GB/s I/O link (to I/O subsystem)

10*2GB/s Intrarack (5-D torus)

** chip I/O shares function with PCI_Express

BGQ Chip architecture



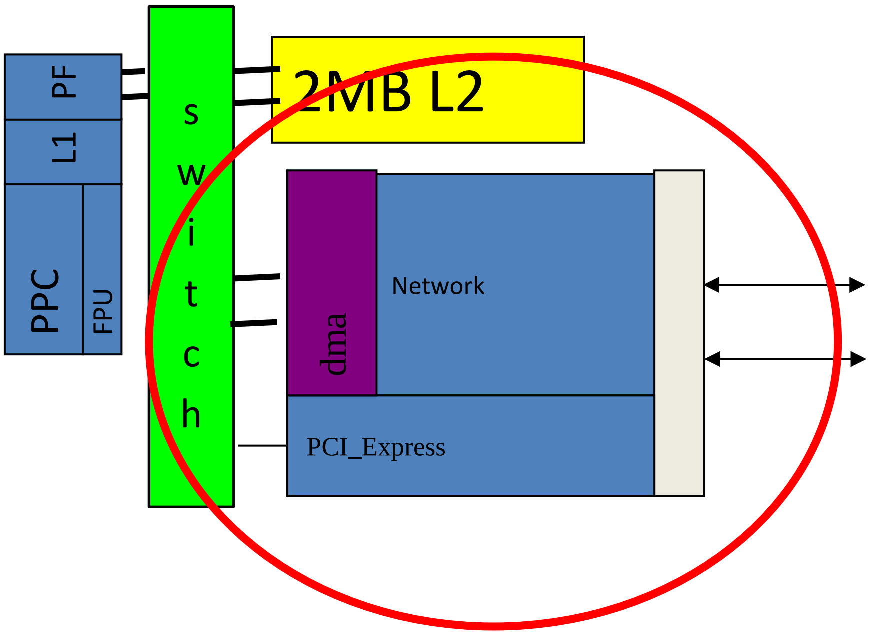
- 16+1 core SMP
  - Each core 4 way hardware threaded
- Transactional memory and thread level speculation
- Quad float point unit on each core
  - 204.8 GF peak node
- Frequency target of (1.6) GHz
- 563 GB/s bisection bandwidth to shared L2
  - (BGL at LLNL has 700 GB/s system bisection)
- 32 MB shared L2 cache
- 42.6 GB/s DDR3 bandwidth
  - (2 channels each with chip kill protection)
- 10 intrarack interprocessor links each at 2.0GB/s
- 1 I/O link at 2.0 GB/s
- 4-8 GB memory/node
- ~30 Watts chip power

2 GB/s I/O link (to I/O subsystem)

10*2GB/s Intrarack (5-D torus)

** chip I/O shares function with PCI_Express

Messaging unit (mu)



- Direct access to L2
- Direct access to network HW
- Fully user programmable
- Performs PtP and collective communications
- Runs independent of cores:
  - Sends data
  - Receives data and stores to memory subsystem
- Shared resource for all 17 cores

Comparison of communication specs

Blue Gene/P	Blue Gene/Q
• Persistent communication: • Inject descriptors once • Reuse as often as possible • Communication startup slow (as slow as non-persistent generic communication) • Communication restart fast: manipulate (memory) Fifo head & rec. and inj. counters • HW: • 4 Fifo groups, 32 Fifos/group • 4 Cntr. groups, 64 couters/group	• Persistent communication • Same setup as for BGP • However: no address associated with reception counter • HW: • 512 Fifos (= 32 fifos/core) • 512 BAT (base address table) entries (= 32 entries/core) • Number of counters "unlimited" →BAT entry required • BAT entry required also for direct put operation.

Mitglied der Helmholtz-Gemeinschaft

LQCD code

- Vanilla (C/MPI), Cuda, BG/Q (extensive use of macros)
- Threaded (pthreads, master/workers)
- Parallelization strategy for BG/Q:
  - 16/4 (AxBxCxDxEx16)(x4)
  - Use shmem window to communicate between processes
  - Synchronize threads/processes with A2 barrier (shmem)
  - Node layout: wrap 2 dimensions into a 4d torus
- Implementation strategy:
  - Use permutes

→ Always stay in 2nd level cache

Mitglied der Helmholtz-Gemeinschaft

BERGISCHE UNIVERSITÄT WUPPERTAL

JÜLICH FORSCHUNGSZENTRUM

Going from P to Q

- In many cases there is a 1 to 1 mapping from BG/P to BG/Q
- Obvious examples:
  - `vec_madd = __fpmadd`
  - `vec_sub = __fpsub`
- Less obvious ones:
  - `vec_xmadd = __fxcpmadd`
  - `vec_xxmadd = __fxcxma`
  - `vec_xxcpnmadd = __fxcxnsma`
  - `vec_xxnpmadd = __fxcxnpma`
- Loads can isolate lower half of register → emulate BG/P (**latency!**)
  - `vec_ld2, vec_st2 = __lfpd, __stfpd`

Treading

- BG/P code was not threaded → 1 core = 1 process
- BG/Q, by default we use 1+3 threads/core
- Started with openMP, but found large latencies
- Moved to pthreads using a master, worker setup
  - → pthreads/worker setup requires rewriting of loops
- pthread barrier too slow, use A2 atomics
- Avoid wasting cycles:
  - Spinning (waiting) workers take cycles from master running serial code
  - → Use Wakeup-Unit: threads sleep and wake up on demand

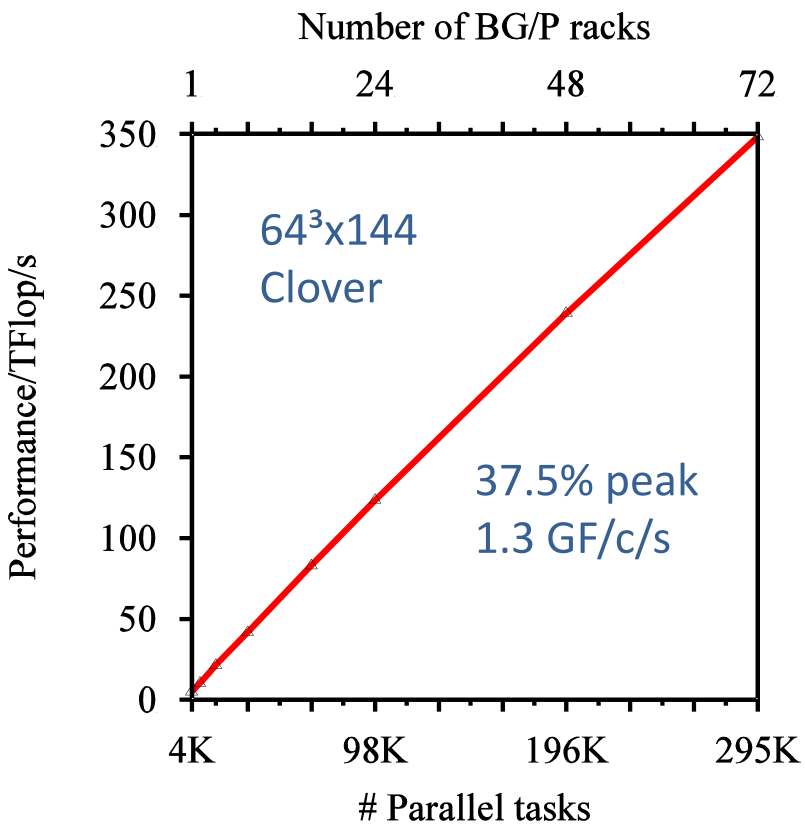
Node-wide barriers – BG/Q L2 atomics

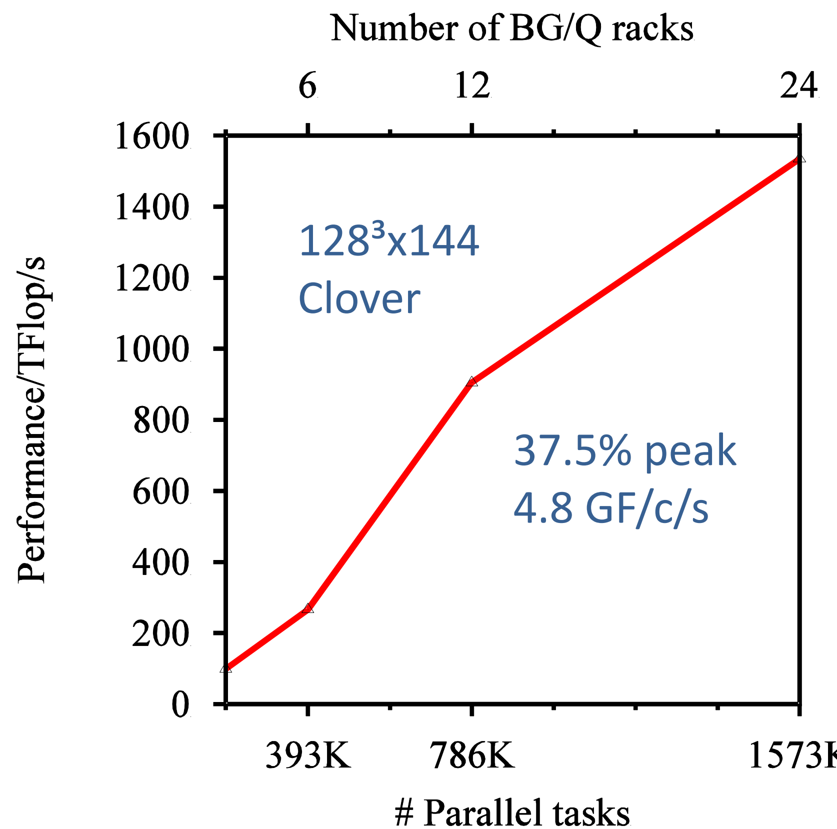
```
__INLINE__ void L2_Barrier(L2_Barrier_t *b, int numthreads)
{
    uint64_t target = b->start + numthreads;
    uint64_t current = L2_AtomicLoadIncrement(&b->count) + 1;

    if (current == target) {
        b->start = current;  // advance to next round
    } else {
        while (b->start < current);  // wait for advance to next round
    }
}
```

Low-level communication, BGP/BGQ

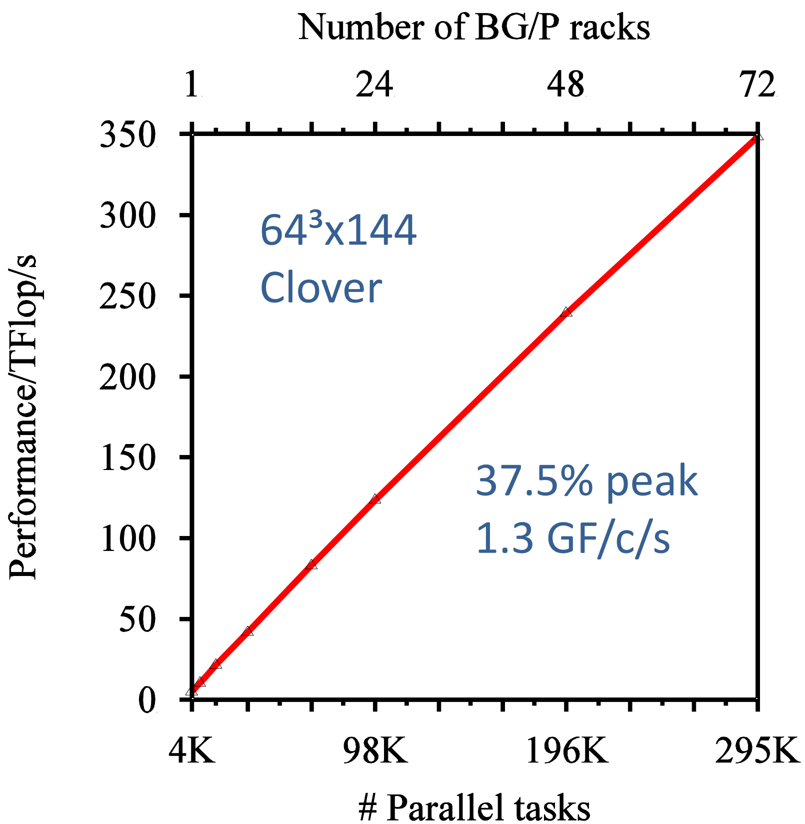
- SPI based
- Code interface requires flexible communication using tags/communicators
- Standard comms are made persistent and are freed in times of need
- Restarts check if the tag/communicator has been force freed, otherwise performs a simple restart
- **Q: SMP (1/64) or multiple process modes (eg. 16/4)**
- Global comms proceed via SMP window (shm_open/mmap)
- **Q: PtP comms contain no global ops (i.e. no global sycs)**

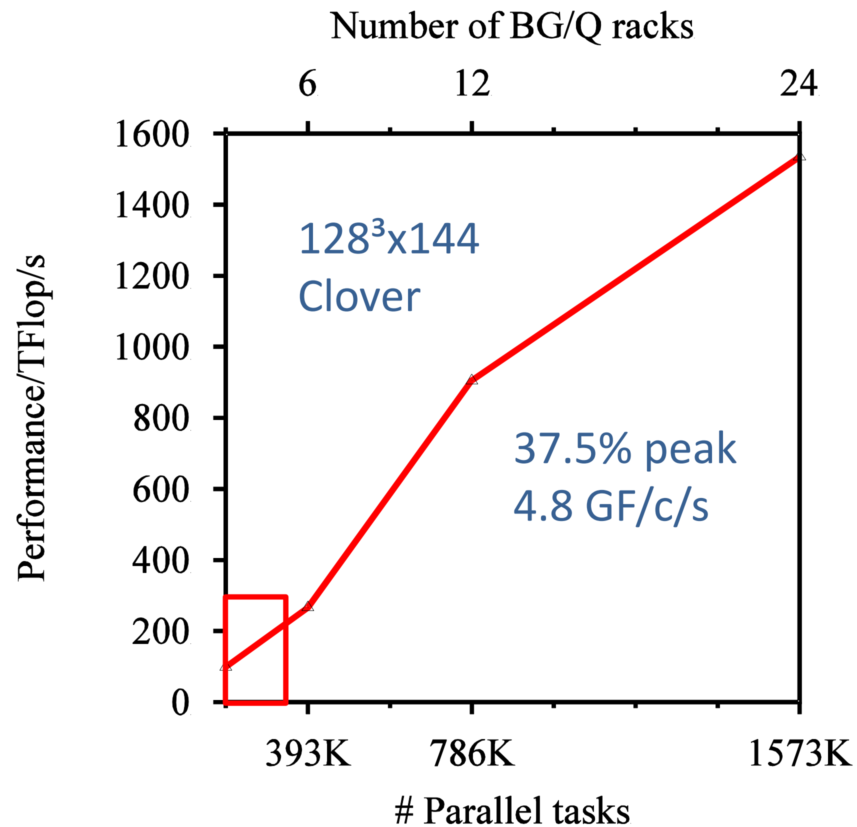
Performance



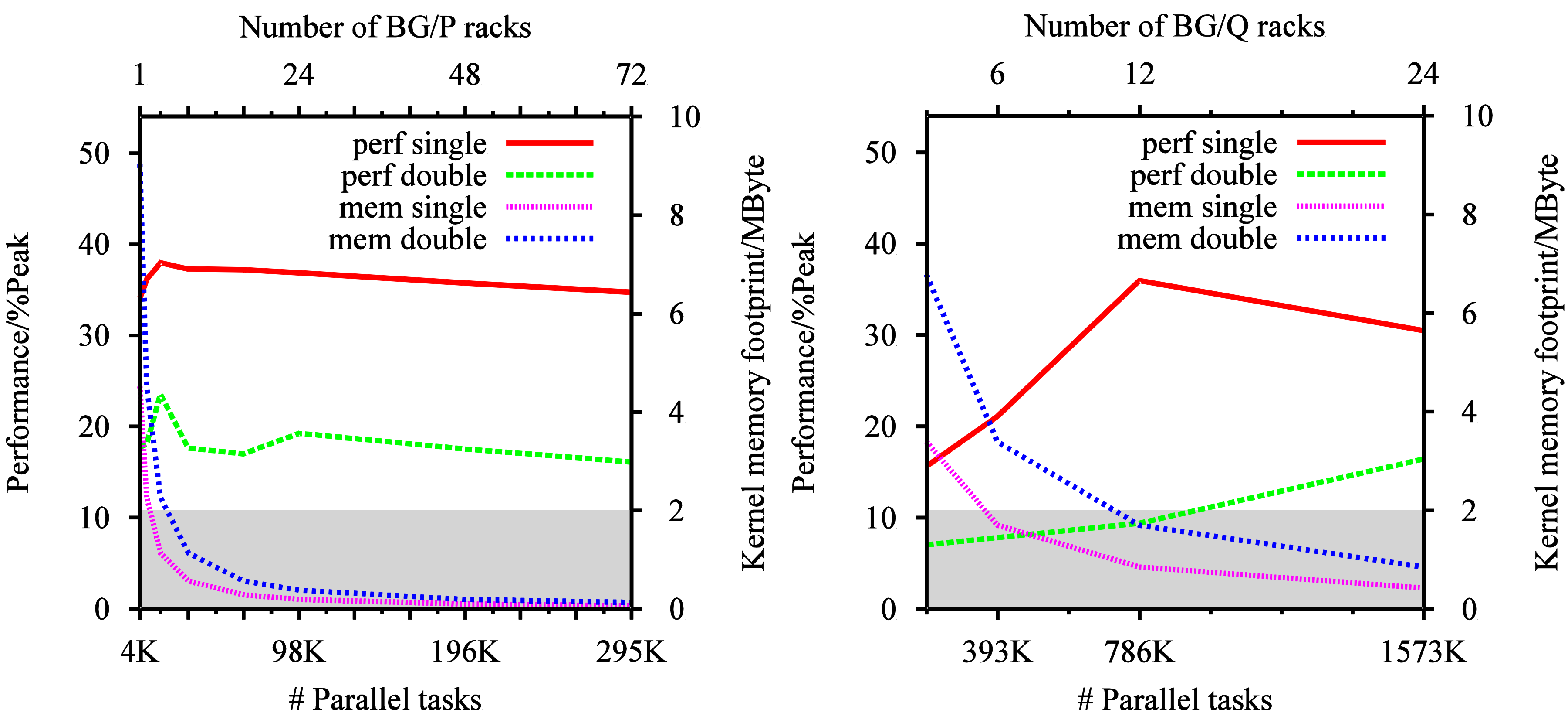


Performance





Performance



Solver performance (production)

- Multishift CG: 3.2 Gflop/core/s (3x6x6x6 loc. lat.) (CG more efficient, BiCGstab similar)
- Multilevel method (A. Frommer, K. Kahl, S.K., B.Leder, and M.Rottmann, 1303.1377, 1307.6101)
  - Smoother: 3.4 Gflop/core/s (12x6x6x3 loc. lat., 48^3x96)
  - Restriction/interpolation: 3.6/6.6 Gflop/s
  - Coarse grid operator: 1.2/2.2 Gflop/s (2x2x1x1 loc. lat.)
  - Total 1.9 Gflop/s
  - Setup: 2.4 Gflop/s

Conclusions

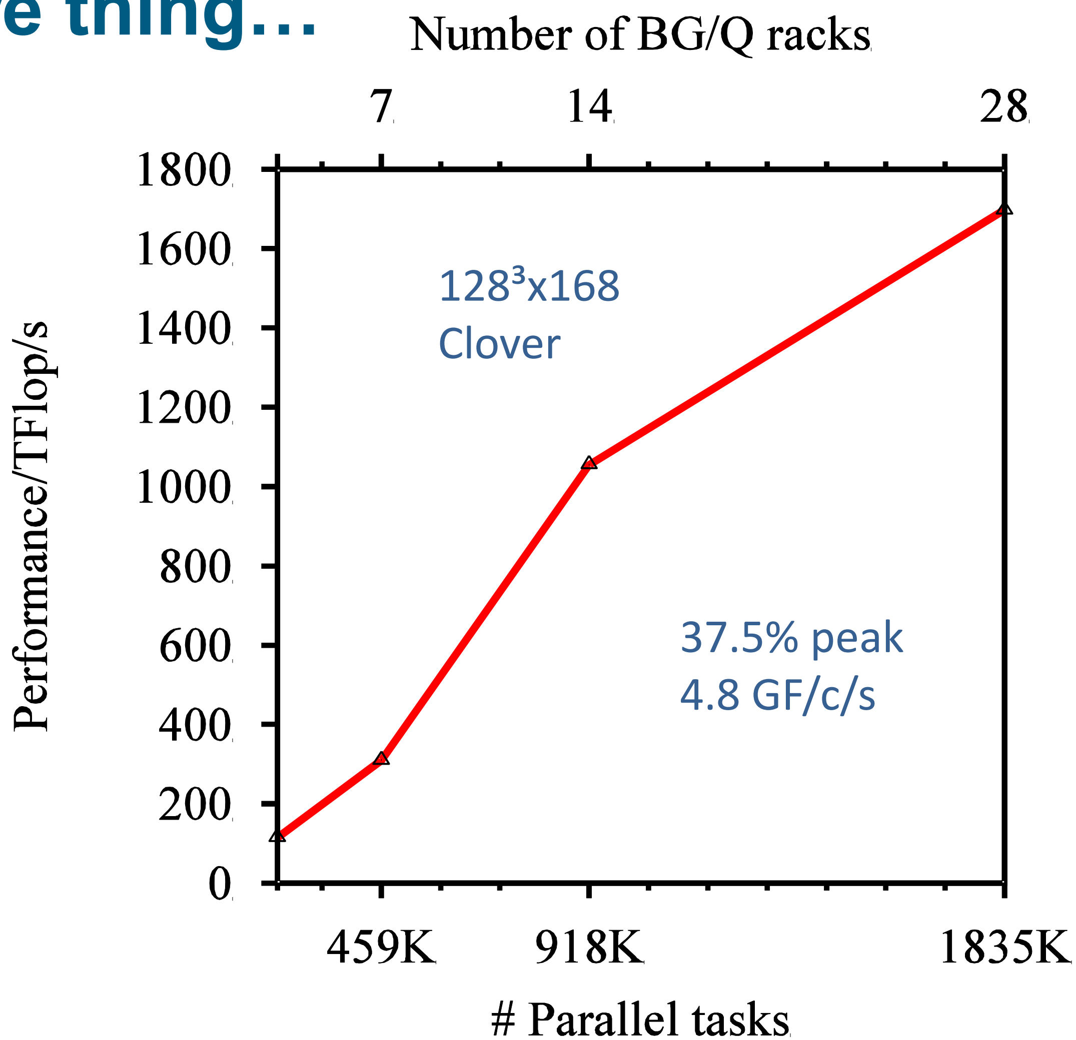
- Efficiency on BG/Q appears to much more peaked when compared to BG/P
- Peak value remains unchanged
- Solver performance noticeably lower than for dslash
- Continuing tendency towards more complex solvers (e.g. MG)

→ Tuning becomes more difficult

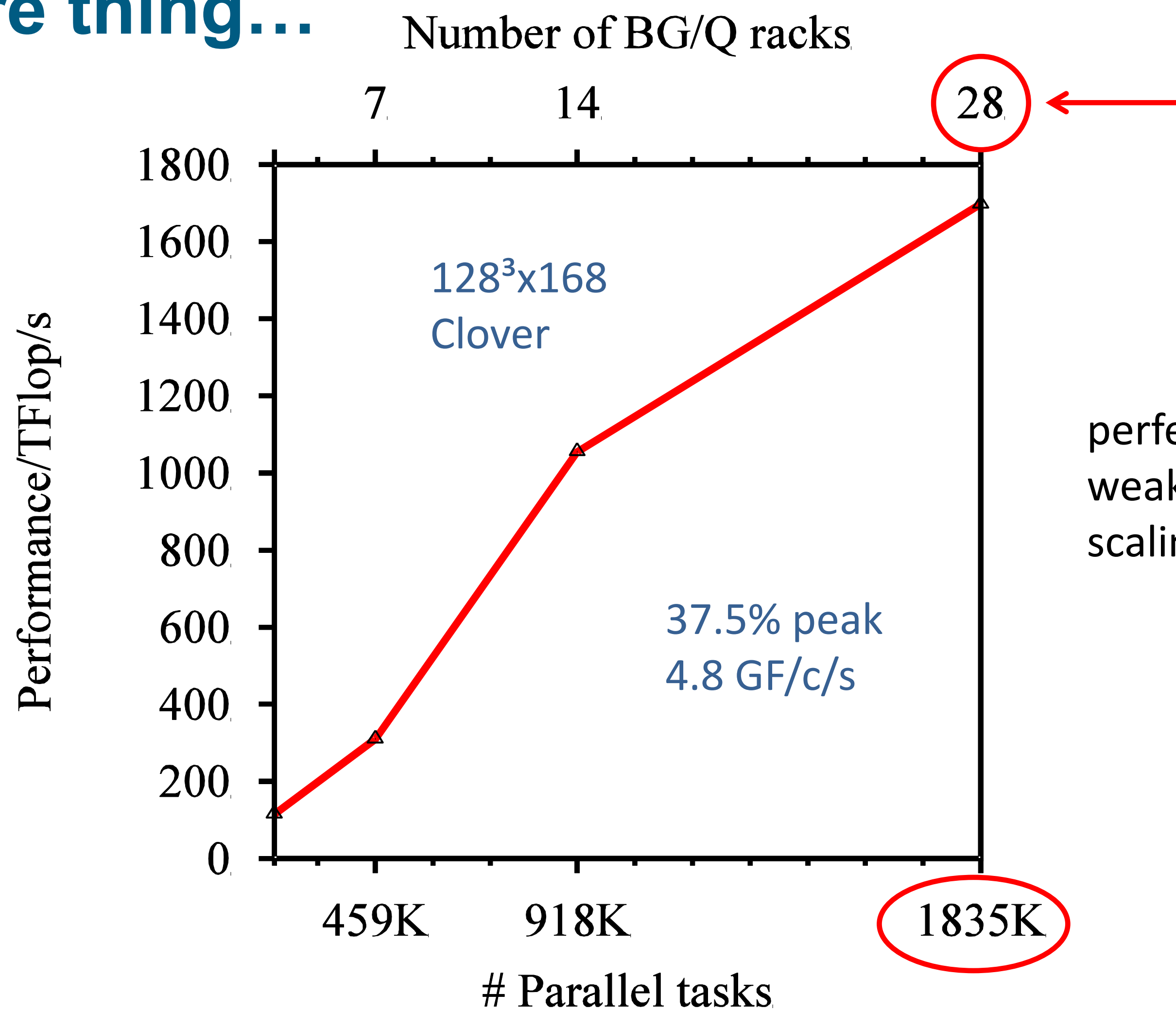
→ So far, performance results of complex solvers do not match those of ordinary ones

→ Full vectorization of code potentially necessary

One more thing…



One more thing…



perfect weak scaling…