Accelerating Twisted Mass LQCD with QPhiX

Mario Schröck†, Silvano Simula†, Alexei Strelchenko‡

†INFN Sezione di Roma Tre

‡Fermi National Accelerator Laboratory

Lattice 2015, Kobe, July 17

MOTIVATION

- the QPhiX library offers high performance on Intel architectures (Xeon and Xeon Phi)
- the newly launched Galileo computer in Cineca, Italy, (one Pflop) makes large use of Xeon Phi accelerators, suitable machine for propagator calculations
- we want to extend QPhiX to our needs for tmLQCD
- can QPhiX be useful for the tmLQCD HMC?

QPhiX

- Original authors:
  Bálint Joó[1], Dhiraj D. Kalamkar[2], Karthikeyan Vaidyanathan[2] *et al.* (2013)
  [1]Jefferson Lab, [2]Parallel Computing Lab, Intel Corporation, Bangalore, India

- QPhiX library (C++) is divided into two parts:

  - code generator abstracts away vector intrinsics (KNC, AVX, AVX2, AVX512, SSE, QPX, scalar)

  - high level part which is concerned with parallelizing over threads (OpenMP), multi-processing (MPI) and performing the loop structure for the cache blocking strategy

- Supports Wilson (Clover) fermions and (mixed prec.) CG, BiCGstab

Twisted Mass Operator

$$\not{D}_{\mathrm{TM}} = \not{D}_{\mathrm{W}} 1_f + i\mu\gamma_5\tau_3$$

- kernels for symmetric even-odd preconditioning:
  - Dslash: $$\chi = R^{-1}\, \not{D}_{\mathrm{W}}\psi$$
  - AChiMBDPsi: $$\phi = A\chi - b\, \not{D}_{\mathrm{W}}\psi$$

    with $$A = R \equiv 1 + i2\kappa\mu\gamma_5$$

    the twist operator.
  - …plus daggered versions.

Twisted Mass Operator

- the successive application of these kernels corresponds to the Schur decomposed operator

$$\tilde{M}_{oo} = R_{oo} - \frac{1}{2}\kappa \not{D}_{oe} R_{ee}^{-1} \not{D}_{eo}$$

- to implement these kernels we mainly have to work on the code-generator

- the high-level part has to be modified to include the twisted mass parameter **μ** and to call the corresponding twisted mass low-level kernels

Dslash

$$\chi = R^{-1}\, \not{D}_{\mathrm{W}} \psi$$

- the routine `dslash_plain_body()` generates the Dslash kernel routines
- within that, after the call of `dslash_body()`, we call the (inverse) twisted term
- this will generate files with vector intrinsics of the form `tmf_dslash_plus_body_float_float_v8_s4_12` for the different floating point precisions, vector lengths, soa-lengths and gaugefield compression types (12 or 18)
- analogously for AChiMBDPsi kernels

Wrapping up kernels

- the generated files are wrapped up as template specializations

```
template<>
inline void
dslash_plus_vec<FPTYPE,VEC,SOA,COMPRESS12>(...)
{
  #include INCLUDE_FILE_VAR(qphix/avx2/generated/
  dslash_plus_body_,FPTYPE,VEC,SOA,COMPRESS_SUFFIX)
}
```

Unpacking routines

- the spinors are projected to halfspinors for the MPI communications
- these packing routines are identical to the Wilson case
- but when *unpacking*, while accumulating the different directions and before streaming to memory, we still have to apply the twisted term
- thus we need our own unpacking routines

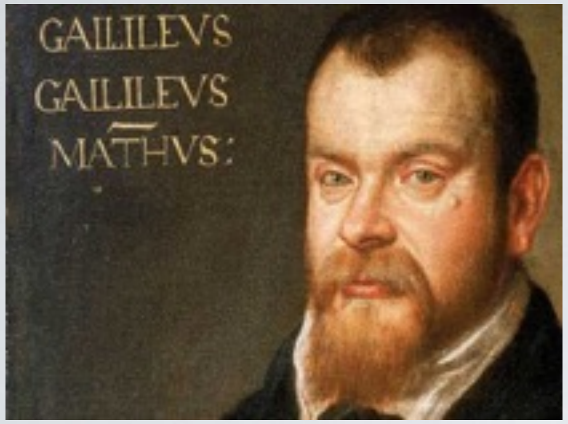
Interfacing with *tmLQCD*

- configure tmLQCD with the options

```
CXX=mpiicpc \
--with-qphixdir=${QPHIXDIR} \
--with-qmpdir=${QMPDIR}
```

- set up the operator in the tmLQCD input file as

```
BeginOperator TMWILSON
  2kappaMu = 0.05
  kappa = 0.177
  Solver = CG
  SolverPrecision = 1e-14
  MaxSolverIterations = 1000
  UseQphixInverter = yes
EndOperator
```

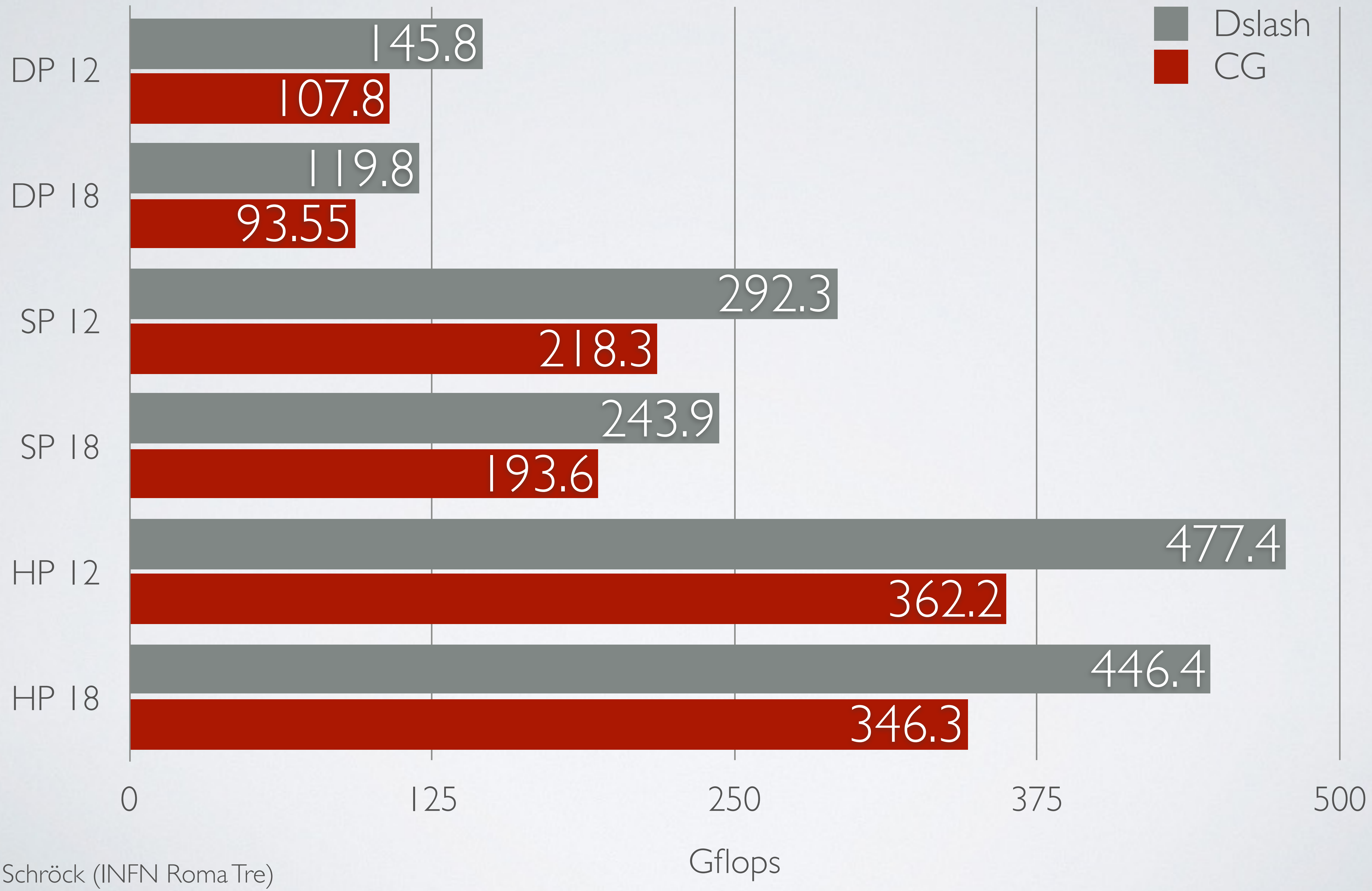


The Galileo Computer

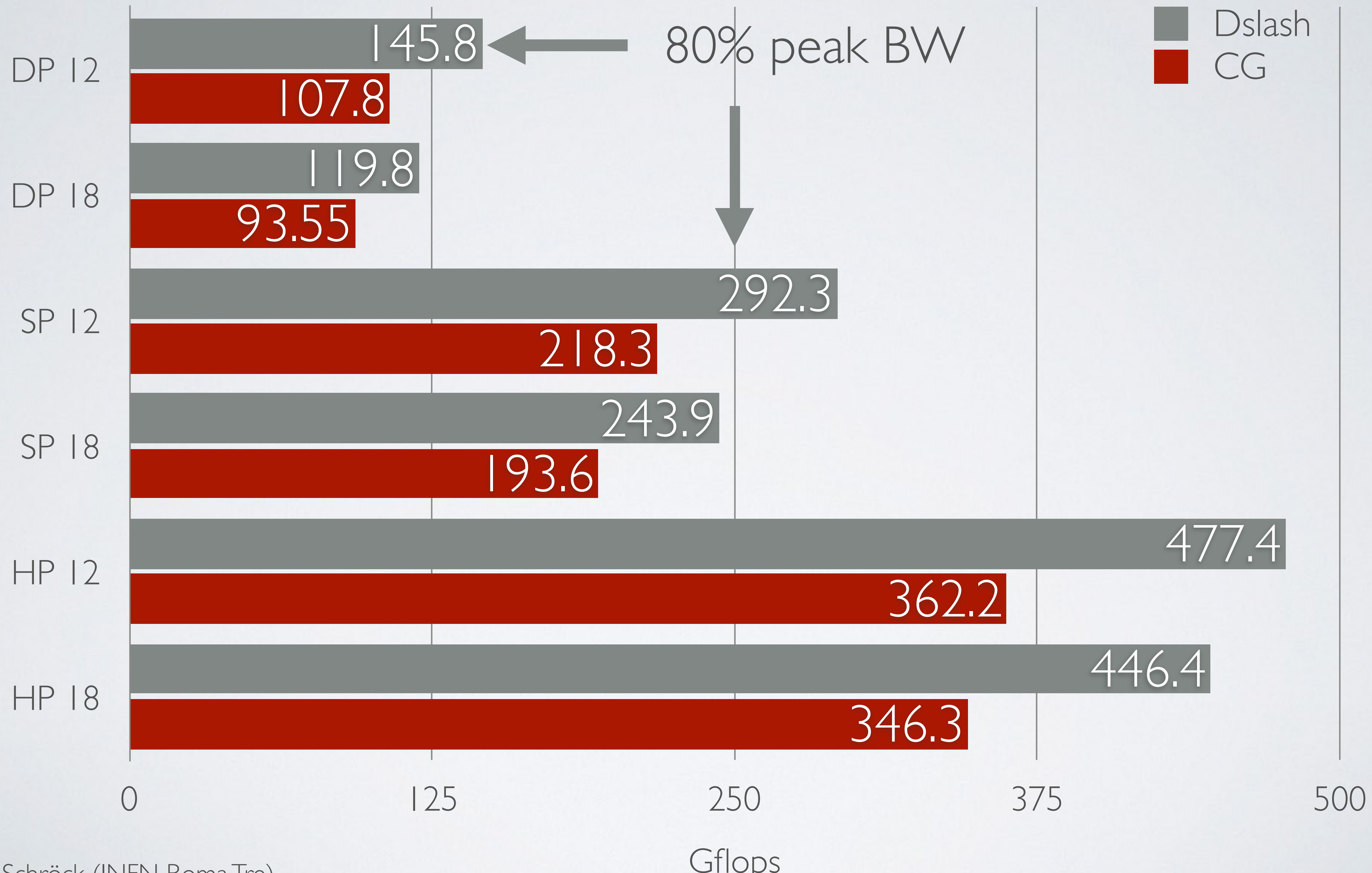
launched in 2015, Cineca, Italy

- 516 compute nodes:
- two octa-core Intel Xeon Haswell CPUs (E5-2630 v3 @ 2.40GHz) per node
- 128 GB RAM per node
- two 16GB Intel Xeon-Phi 7120P (MIC) on 384 nodes
- two 24GB NVIDIA K80 GPUs on 40 nodes
- $\gtrsim$ one Petaflop

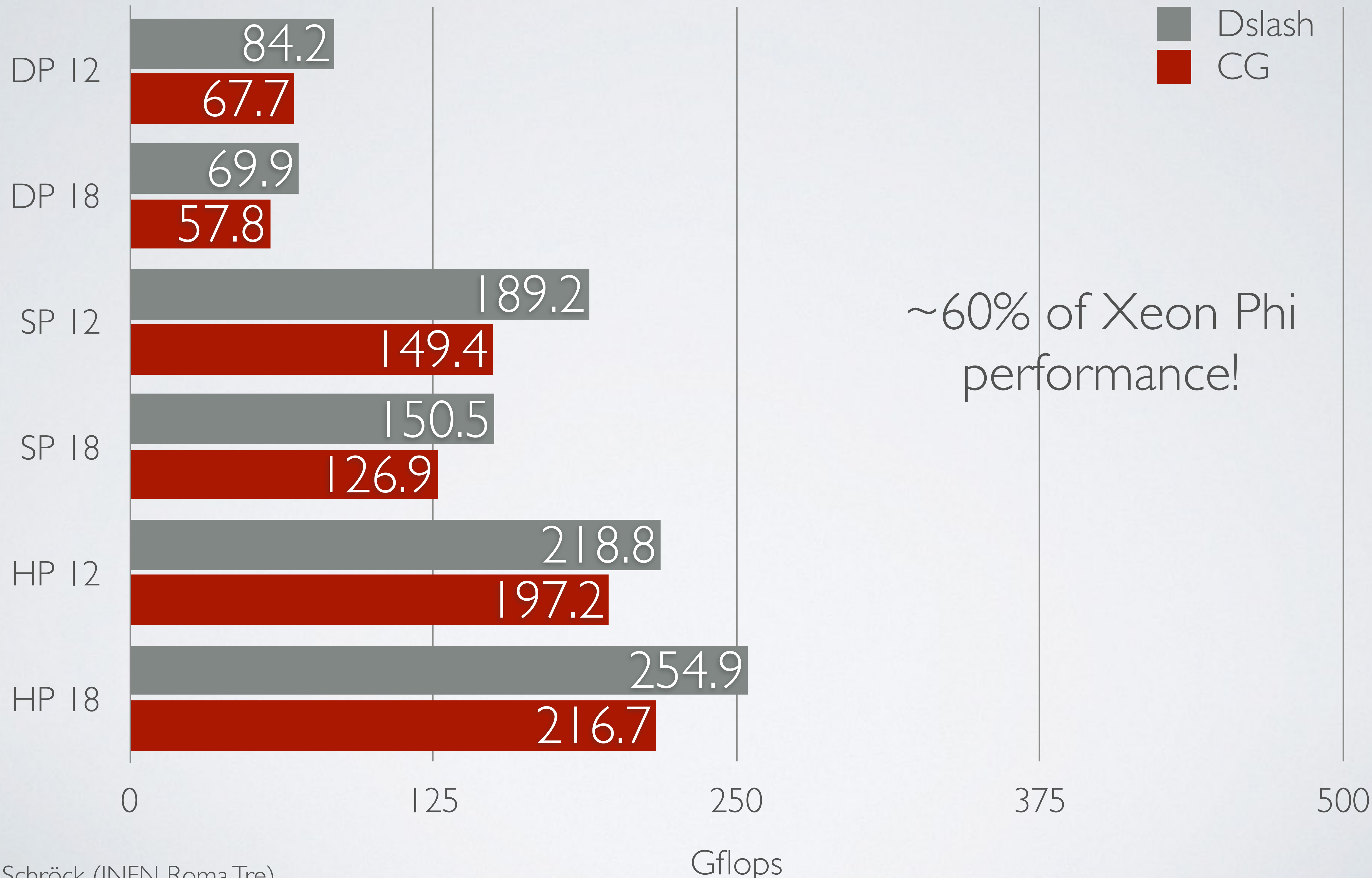
Twisted Mass 32^3x64
Xeon-Phi 7120P



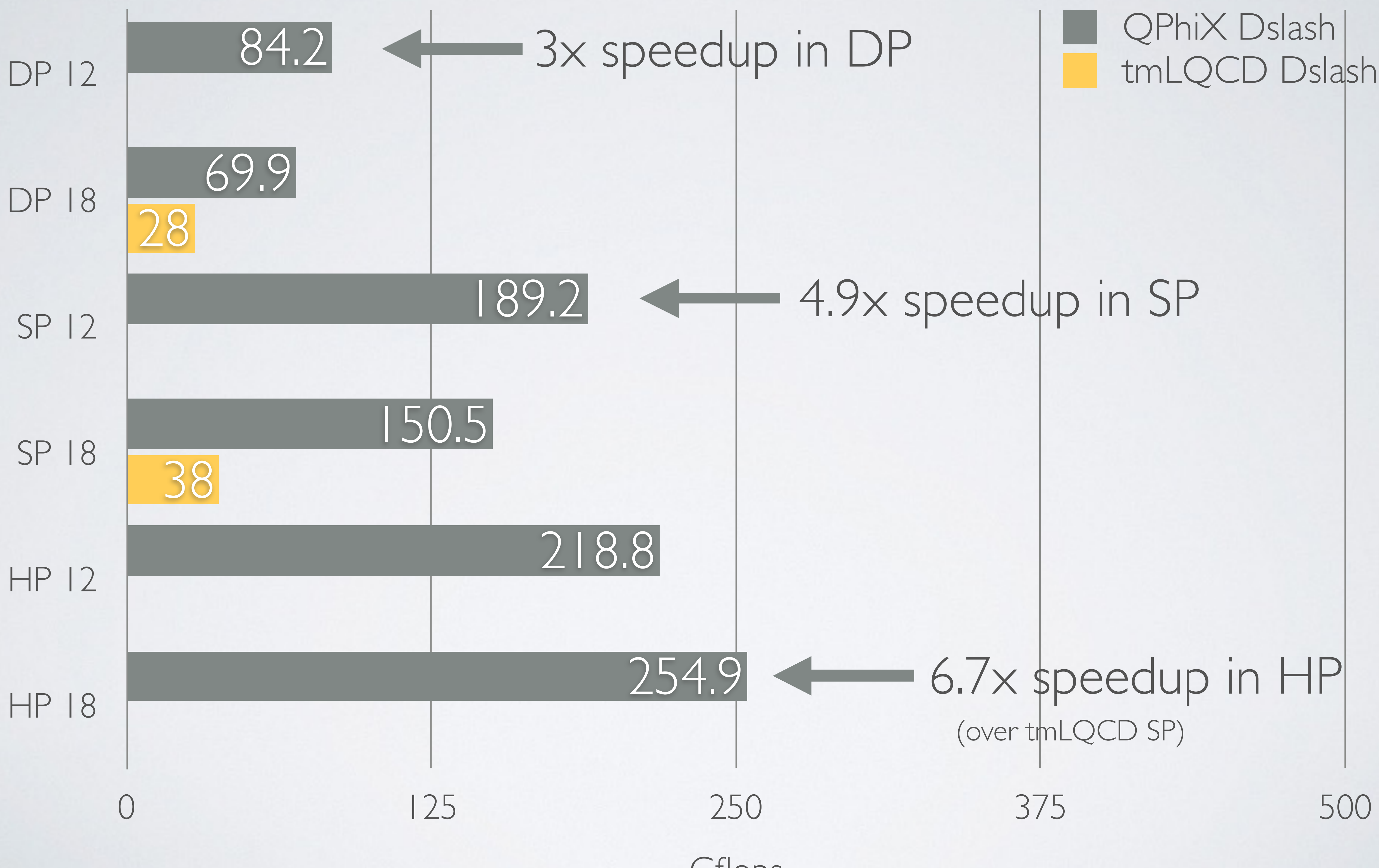
Twisted Mass 32^3x64
Xeon-Phi 7120P



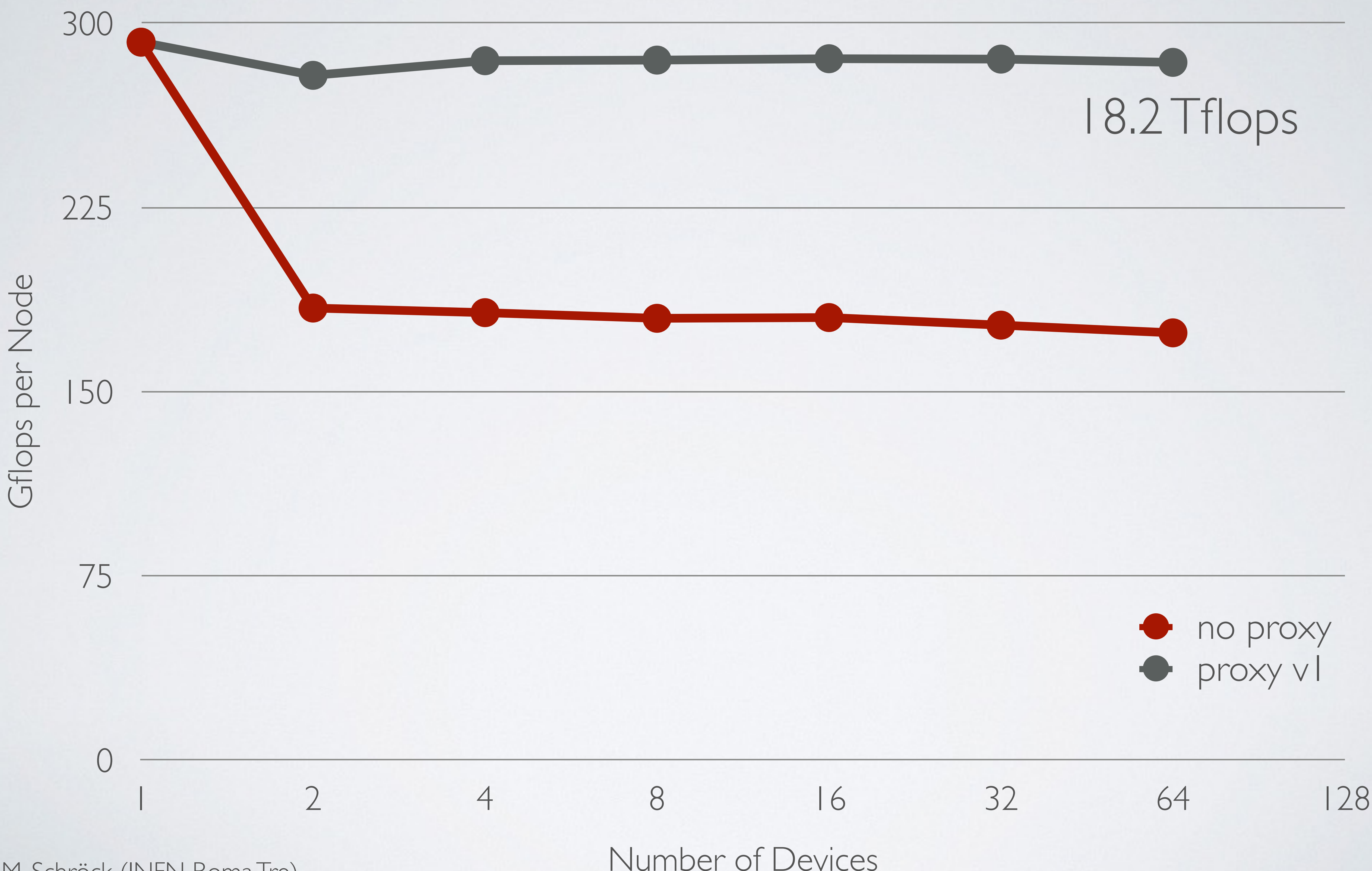
Twisted Mass $32^3 \times 64$
Dual Socket Haswell E5-2630 2.4GHz AVX2



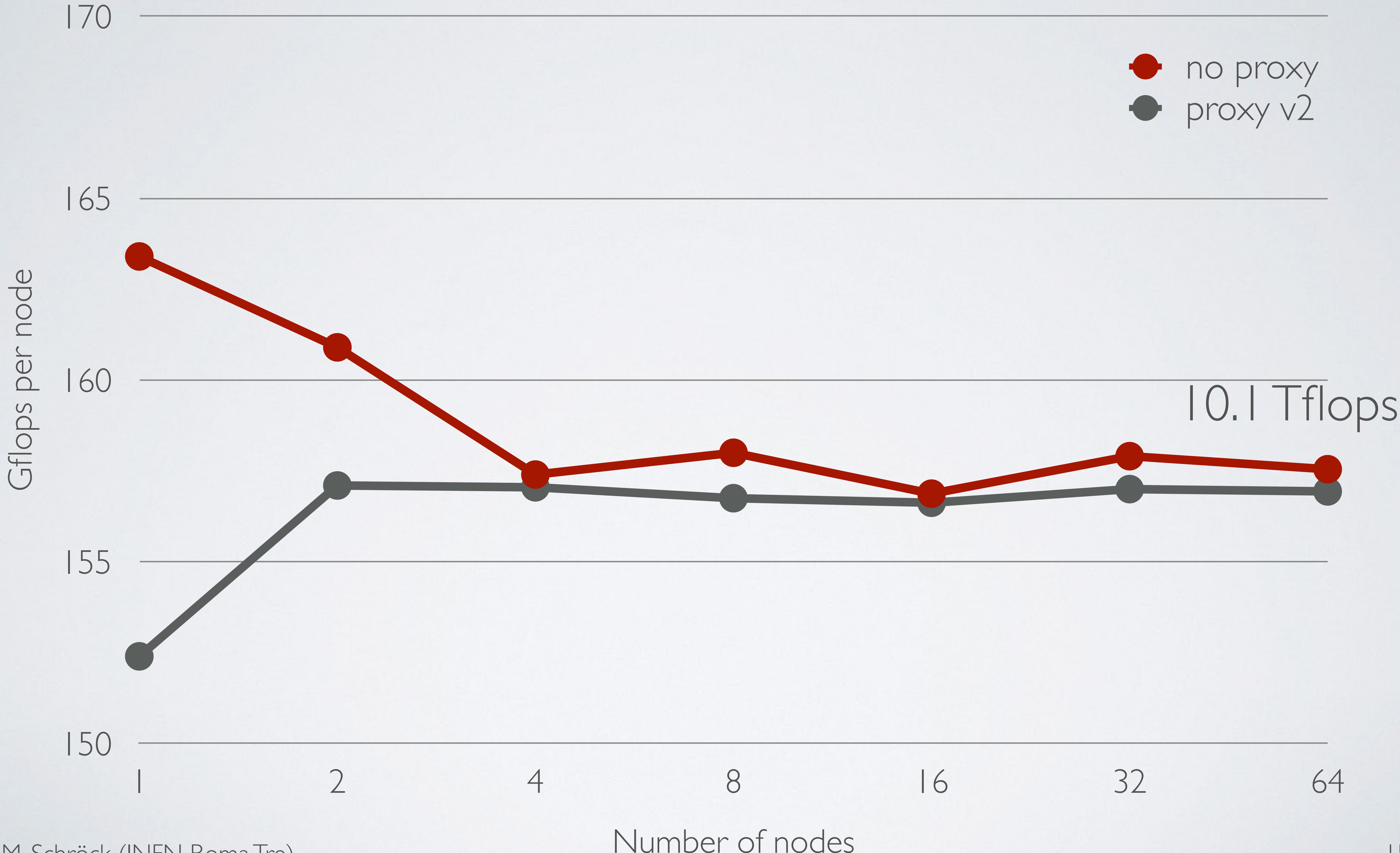
Twisted Mass 32^3x64
Dual Socket Haswell E5-2630 2.4GHz AVX2



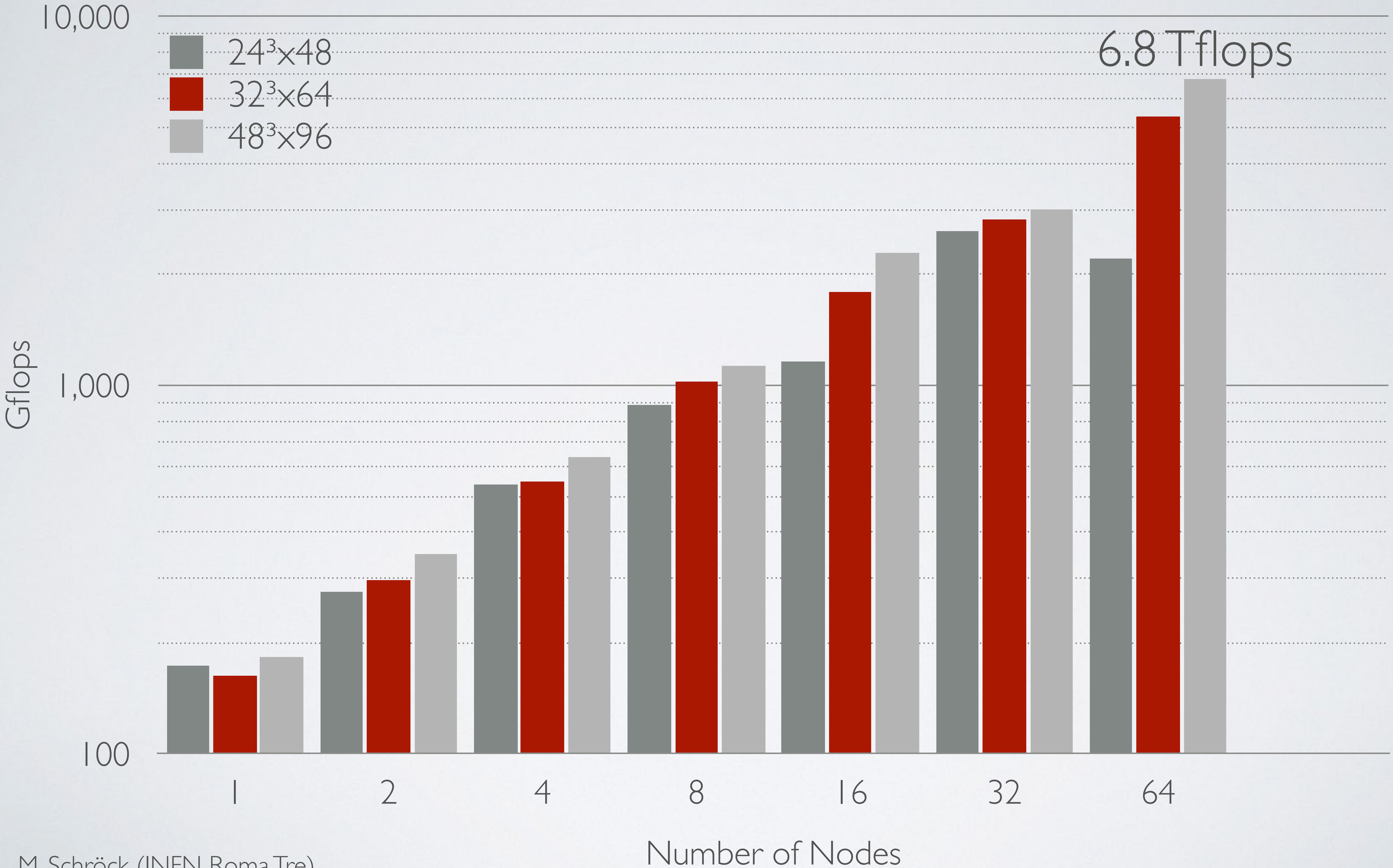
Weak Scaling Twisted Mass Dslash SP 48^3x96 per device Xeon-Phi 7120P



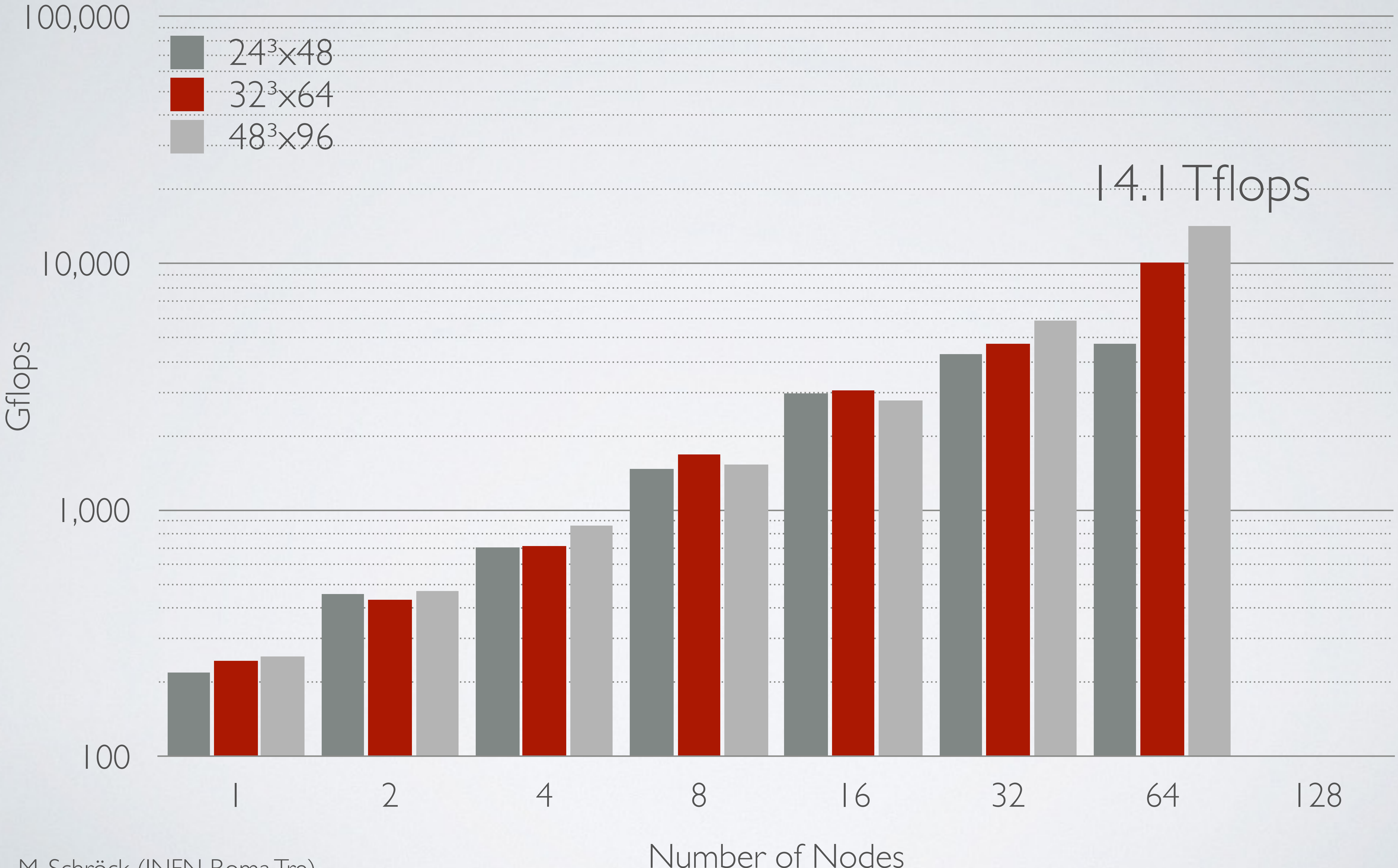
Weak Scaling Twisted Mass Dslash SP 32^3x64 per node
Dual Socket Haswell E5-2630 2.4GHz AVX2



Strong Scaling Twisted Mass Dslash SP Compression12
Dual Socket Haswell E5-2630 2.4GHz AVX2



Strong Scaling Twisted Mass Dslash HP Compression18
Dual Socket Haswell E5-2630 2.4GHz AVX2



OUTLOOK

- first tests with TM QPhiX are very promising, Dslash single prec. performance ~5x higher than tmLQCD
- start production runs (propagator calculations) soon
- implement Clover for TM and non-degenerate doublet - this would allow running the tmLQCD HMC with QPhiX - strong scaling is good!

Thanks for your attention.

We acknowledge assistance with QPhiX by
Bálint Joó,
Dhiraj D. Kalamkar,
Karthikeyan Vaidyanathan.