cuLGT: Lattice Gauge Fixing on GPUs

Hannes Vogt*, Mario Schröck†
*Universität Tübingen, †INFN Roma Tre

Motivation

- QCD action is gauge invariant
- fixing a gauge is necessary in the continuum
- gauge-variant two point functions play an important role in confinement scenarios
- on the lattice:
  - high dimensional optimization problem
  - many local maxima: Gribov copies
  - algorithms are nicely parallelizable
- cuLGT
  - CUDA-based code for gauge fixing and more
  - code is publicly available

Landau gauge on the lattice

- Action is invariant under gauge transformation of the links

$$U_\mu(x) \to U^g_\mu(x) = g(x)U_\mu(x)g^\dagger(x+\hat{\mu}), \quad g(x), U_\mu(x) \in SU(3)$$

- Landau gauge (continuum: $\partial_\mu A_\mu(x) = 0$) is achieved by maximizing

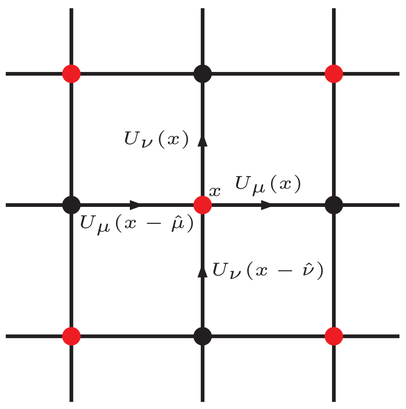
$$F^U[g] = \frac{1}{N_d N_c V} \sum_x \sum_\mu \text{Re tr}\left[g(x)U_\mu(x)g^\dagger(x+\hat{\mu})\right]$$

- (over)relaxation: optimize locally

$$g(x)K(x) = g(x)\sum_\mu \left[U_\mu(x)\overbrace{g(x+\hat{\mu})^\dagger}^{\mathbb{1}} + U_\mu(x-\hat{\mu})^\dagger \overbrace{g(x-\hat{\mu})^\dagger}^{\mathbb{1}}\right]$$

- SU(2): optimum for $g(x) = K^\dagger(x)/\det(K^\dagger)$
  for $N_c > 2$ iterate over SU(2) subgroups

The algorithm: outline



update:

- $U_\mu(x) \to g(x)U_\mu(x)$
- $U_\mu(x-\hat{\mu}) \to U_\mu(x-\hat{\mu})g(x)^\dagger$
- ...

iterate until:

$$\theta \approx \max_x |\partial_\mu A_\mu(x)| < \epsilon$$

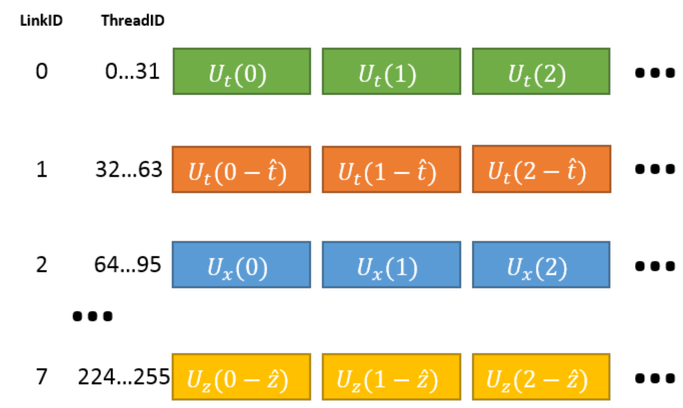
The algorithm: pseudocode

while precision θ not reached **do**
  for sublattice = even, odd **do**
    for all x of sublattice **do**
      for all SU(2) subgroups **do**
        local optimization: find $g(x) \in$ SU(2)
        which is a function of $U_\mu(x),\ U_\mu(x-\hat{\mu})$
        for all μ **do**
          apply $g(x)$ to $U_\mu(x),\ U_\mu(x-\hat{\mu})$

Standard optimizations

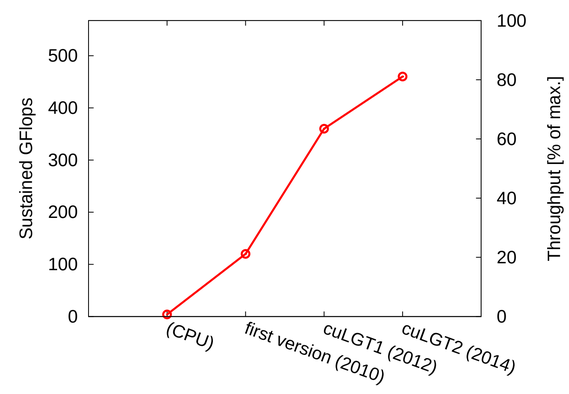
- reorder memory layout for coalescing: memory split into even/odd sublattice; site index running faster than color/Dirac indices[1].
- code is memory bound: store only parts of the $N_c \times N_c$ matrix in global memory. Reconstruct in registers using SU(N_c) properties, for SU(3): 12 parameters instead of 18.
- use texture cache to load links

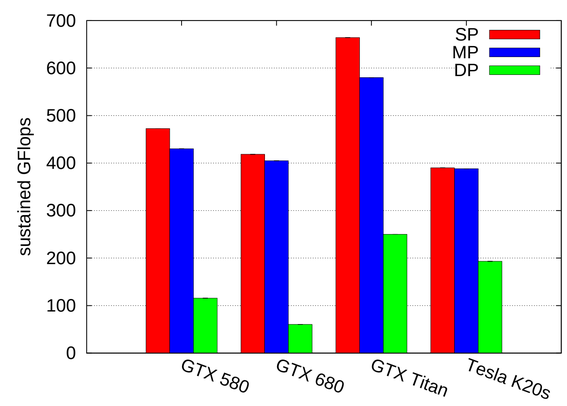
Reduce register spillings: eight threads/site



- At each site: 8 links * 18 parameters = 144 floats.
- Fermi: only 63 registers/kernel. Reduce register usage: with 8 threads per site only 18 floats (registers) for the link (in practice: 34 registers used, no spilling) [2].

Performance





cuLGT2 sample application

```
typedef LocalLink<SUNRealFull<3,float> > LOCALLINK;
typedef SU3Vector4<float> PARAMTYPE;
typedef GPUPattern<SITETYPE,PARAMTYPE> PATTERNTYPE;
typedef GLOBLINK<PATTERNTYPE,USETEXTURE> GLOBLINK;

__global__ polyakovLoop( float4* U, float* polyakov, LatDim<4> dim )
{
  SITETYPE site( blockIdx.x * blockDim.x + threadIdx.x );
  LOCALLINK linkProduct; linkProduct.identity();
  for( int t = 0; t < dim.getDimension( TDIR ); t++ )
  {
    GLOBLINK glob( U, site, TDIR );
    LOCALLINK link = glob;
    linkProduct *= link;
    site.setNeighbour( TDIR );
  }
  polyakov[site.getIndex()] = linkProduct.retrace();
}

int main()
{
  LatDim<4> dim( Nt, Nx, Ny, Nz );
  GaugeConfiguration<PATTERN> config( dim );
  config.allocateMemory();
  // allocate dPolyakov, load the configuration to host memory
  config.copyToDevice();

  GLOBLINK.bindTexture(config.getDevicePointer(),config.getSize());
  polyakovLoop<<<GRIDSIZE,BLOCKSIZE>>>( config.getDevicePointer(),
     dPolyakov, dim );

  reduction( dPolyakov );
}
```

Summary: what does cuLGT offer?

- gauge types: Landau gauge, Coulomb gauge, Maximally Abelian gauge
- algorithms: Overrelaxation, Simulated Annealing
- gauge groups: SU(2), SU(3) (SU(N) easy to implement)
- multi-GPU: only Landau gauge
- integration in other frameworks: MILC
- autotune utility selects optimal setup for different architectures

References

[1] M. Clark *et al*, *Comput.Phys.Commun.***181** (2010) 1517. arXiv:0911.3191.
[2] M. Schröck and H. Vogt, *Comput.Phys.Commun.***184** (2013) 1907. arXiv:1212.5221.

Download

http://www.cuLGT.com/