Rács QCD GPU-kon

Katz Sándor

ELTE, Elméleti Fizikai Tanszék

GPU nap, RMKI, 2010. június 4.

Outline

1. Introduction
2. The ELTE GPU cluster
3. QCD on GPU's
4. Physics results
5. Summary

Introduction

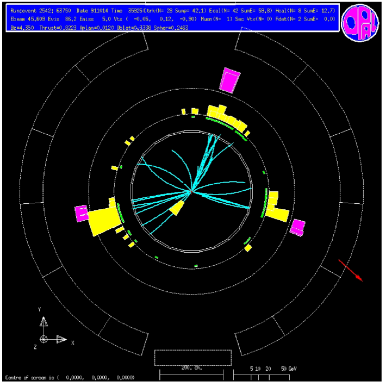
Quantum Chromodynamics(QCD) is the theory of the strong interaction
it confines quarks and gluons inside the proton

QCD is similar to Quantum Electrodynamics with
more complicated symmetry (SU(3) instead of U(1))
quarks have three ”colors”

At high temperatures hadrons break up
and the quark-gluon plasma is formed

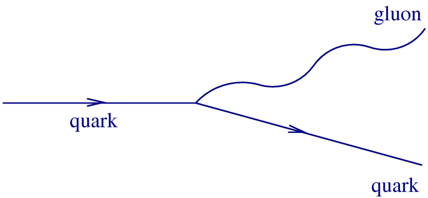
Some interesting questions
At what temperature does the transition happen?
What is the equation of state of the quark-gluon plasma?
What happens at non-zero baryon density?

Basic interaction(s) in QCD



```
Run:event 2542; 63750  Date 911014 Time  35925 Ctrk(N= 28 Sump= 42.1) Ecal(N= 42 SumE= 59.8) Hcal(N= 8 SumE= 12.7)
Ebeam 45.609 Evis  86.2 Emiss   5.0 Vtx (  -0.05,   0.12,  -0.90) Muon(N=  1) Sec Vtx(N= 0) Fdet(N= 2 SumE=  0.0)
Bz=4.350  Thrust=0.8223 Aplan=0.0120 Oblat=0.3338 Spher=0.2463
```

quark emits a gluon
(or a gluon emits one/two gluons)



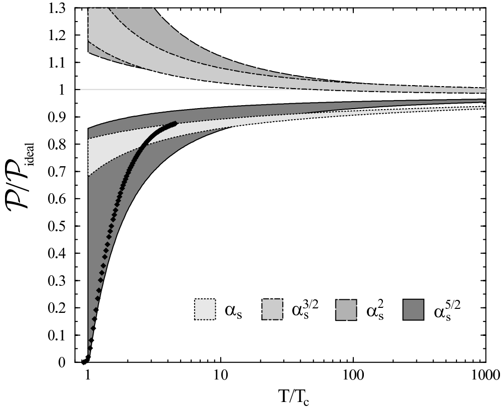
at LEP the process can be clearly seen (≈10% of QCD processes)

we see jets and varify the underlying equations: asymptotic freedom
we do not see free quarks or gluons: confinement phenomena

QCD: need for a systematic non-perturbative method

in some cases: good perturbative convergence; in other cases: bad

pressure at high temperatures converges at T=10^{300} MeV



Lattice field theory

systematic non-perturbative approach (numerical solution):

quantum fields on the lattice

quantum theory: path integral formulation with S=E_{kin}-E_{pot}

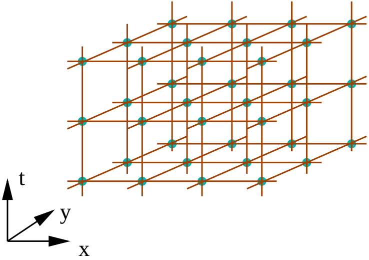
quantum mechanics: for all possible paths add exp(iS)
quantum fields: for all possible field configurations add exp(iS)

Euclidean space-time (t=$i\tau$): exp(-S) sum of Boltzmann factors

we do not have infinitely large computers ⇒ two consequences

a. put it on a space-time grid (proper approach: asymptotic freedom)
formally: four-dimensional statistical system
b. finite size of the system (can be also controlled)

⇒ stochastic approach, with reasonable spacing/size: solvable

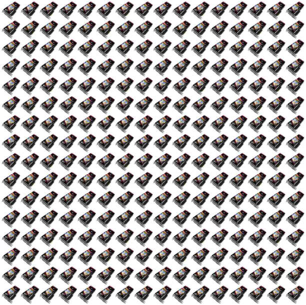


fine lattice to resolve the structure of the proton ($\lesssim$0.1 fm)
few fm size is needed
50-100 points in 'xyzt' directions
$a \Rightarrow a/2$ means 100-200×CPU

mathematically
10^9 dimensional integrals

advanced techniques,
good balance and
several Tflops are needed

The GPU cluster at ELTE



Special hardware:

graphics cards (GPU's) → 120 GFlop sustained /card
128 nodes / 256 GPU's → ≈ 30 TFlop
ideal for lattice calculations

Cluster details

128 nodes (160 by the end of July):

intel corei7 CPU, 2.67 GHz
12 Gbytes RAM
500 Gbytes HDD
2x NVIDIA gtx275

Interconnect

40 Gbit/s infiniband
36 port switches (32 nodes)

Network performance (full duplex)

$\approx$80 Gbit/s between two cards/node
$\approx$55 Gbit/s between nodes
latency: $\approx 3\mu$s

Previous experience with clusters

1998

32 node PC cluster (one of the first for lattice)
AMD K6/2 300MHz CPUs
3dNow MMX instructions utilized
later extended to 96 nodes

2001

128 node PC cluster with gigabit Ethernet
SSE instructions
First machine to hit $1/Mflops threshold

Since 2005

GPU solutions (first for lattice)
OpenGL (Cg) then CUDA
several GPU generations:
7800-7900GTX, 8800GTX, GTX260-275 (GTX480 is coming)

QCD on GPU's

We need to solve

$$Dx = b,$$

where D is the discretized Dirac-operator, a sparse matrix

Iterative solution, necessary ingredients

Dx matrix-vector multiplication

$y = ax + b$, etc. linear algebra with complex coefficients

$r = x \cdot y$ scalar product → global sums

host ↔ device transfer ”slow” → entire solver on the GPU

1. Upload D and b to the device
2. Solve on the GPU
3. Download x

Structure of D

D connects neighboring sites, its "elements" are SU(3) matrices.
E.g. for staggered fermions:

$$(D\Psi)_x = m_q \Psi_x + \frac{1}{2} \sum_{\mu=1..4} \alpha_{x\mu} \left(U_{x\mu} \Psi_{x+\hat{\mu}} - U^{\dagger}_{(x-\mu)\mu} \Psi_{x-\hat{\mu}} \right),$$

where $\alpha_{x\mu} = \pm 1$ and $U_{x\mu}$ are SU(3) matrices.

Do not store matrix elements of D in GPU memory,
but instead store Ψ_x and $U_{x\mu}$ and pointers to neighbors
Each U is a 3x3 complex matrix $\rightarrow$ 18 real numbers
Unitarity: only 8 parameters would be needed (possible, but unstable)
optimization: store only 2 rows of U, third can be easily reconstructed
reduces the total required memory and # of memory accesses

CUDA implementation

– one thread per lattice site

– store $U_{x\mu}$, Ψ_x and neighbor tables in registers and shared memory
both are needed to allow large enough block size

– block size is limited by register and shared memory usage
typically set to 64
constraint on lattice extensions → can be avoided by padding

– grid size is determined by lattice size
can accommodate up to 32^4 lattices on a single GPU

– global sums by parallel reduction
only down to one number/block, rest is done on CPU

Parallelization

– D is local → split up lattice to smaller subvolumes

– only Ψ needs to be communicated and only on the surface
required communication bandwidth is $\approx$ 2 orders of magnitude
smaller than the memory bandwidth → still O(10) Gbit/s is needed

– asynchronous device-host transfers →
part of the communication can be hidden by computations

– MPI is used for inter-process communication
one process per GPU
alternatively openMP could also be used for the two cards/node

– communication loss is $\lesssim$30% for up to 4 GPU-s on a $24^3 \cdot 64$ lattice

Precision

most current GPU's have limited double precision support
What if we need the solution in double precision?

Multiprecision solvers

- most computations in single (32bit) precision
- few iterations in double precision
- result is correct up to double precision
- even half (16bit) precision can be used
- cannot be used without limits as # of iterations increases

Simple example

1. Solve $Dx = b$ in single precision.
2. Evaluate $Dx - b$ in double precision

$$Dx - b = r$$

where $|r|/|b| < 10^{-6}$

3. Solve again for r

   solution of $Dx' = r \rightarrow Dx' - r = r'$

   with $|r'|/|r| < 10^{-6}$

4.Combine the results:

$D(x - x') - b = -r'$, so $x - x'$ is accurate up to 10^{-12}

Performance

One node performance

best on large lattices (contrary to CPU codes)
staggered fermions: up to 90 Gflops
Wilson fermions: up to 120 Gflops

Parallel performance

so far up to 4 GPU's (soon to be extended to 64 GPU's)
≈300 GFlop can be reached on 4 GPU's (two nodes)

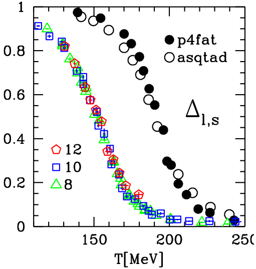
Comparison:

1 BlueGene/P rack with highly optimized code: ≈ 5 Tflops
costs around $1 million
assuming scaling: equivalent to 64 GPU's, 32 PC's !

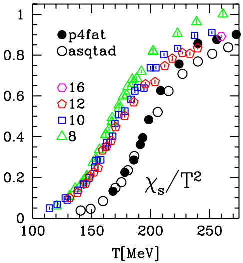
All numbers are sustained performances, peak is much higher

Transition temperatures for various observables

chiral condensate

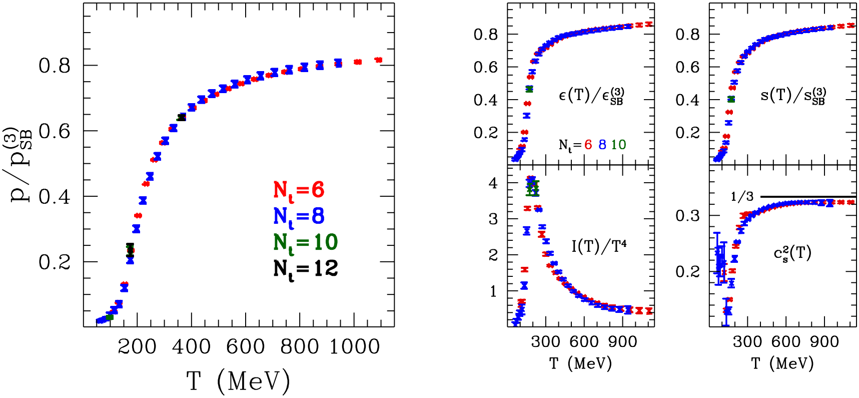


quark number susceptibility



	$\chi_{\bar\psi\psi}/T^4$	$\chi_{\bar\psi\psi}/T^2$	$\chi_{\bar\psi\psi}$	$\Delta_{l,s}$	L	χ_s
WB'09	146(2)(3)	152(3)(3)	157(3)(3)	155(2)(3)	170(4)(3)	169(3)(3)
WB'06	151(3)(3)	-	-	-	176(3)(4)	175(2)(4)
BBCR	-	192(4)(7)	-	-	192(4)(7)	-

Equation of state

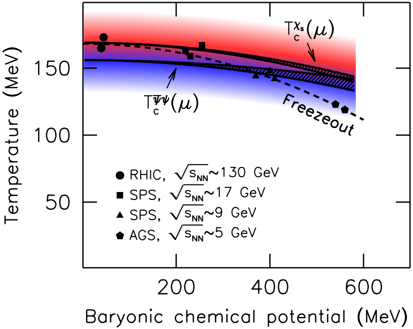


- Two lattice spacings ($N_t = 6, 8$) + checkpoints ($N_t = 10, 12$)
- nice scaling
- everything is derived from the pressure

The QCD phase diagram

non-zero chemical potential → sign problem
Monte-Carlo based on importance sampling fails

We can still calculate derivatives at $\mu = 0$
Phase diagram for relatively small μ can be given



Summary

- GPU's are optimal for lattice QCD calculations
- the 256 GPU cluster at ELTE has $\approx$ 30 Tflops sustained performance
- CUDA implementation with efficient parallelization is possible
- Multiprecision solvers reduce the need for double precision operations
- Large scale simulations on GPU's produce important physics results