Non commutative geometry on GRID

Spisso Bernardino[1,2]

[1]University of Naples Federico II
Napoli,Italy

[2]INFN-Napoli Section -
Italy

20/05/2014-EGI Community forum

So, what is Noncommutative Geometry?

The short answer is: a theory for which there is not commutative rule $[x, y] = xy - yx \neq 0$

I prefer to say see non commutative geometry as a set of tools which describes geometry not as a set of points, lines, vectors etc. but rather using the functions defined on it. And the tools are sufficiently flexible to be applied also in cases for which it does not make sense to talk of points of the space.

It is the direct extension of the quantum mechanics ideas ruled by the Heisenberg principle:

$$[x, p] = ih \tag{1}$$

Several of the recent applications of non commutative geometry concern the possibility that space time at the Planck length is described by a different form of geometry

In order to measure the position of an object, and hence the point in space, one has use a very small probe, and quantum mechanics forces us to have it very energetic, but on the other side general relativity tells us that if too much energy is concentrated in a region a black hole is formed.

This has led to the study of field and gauge theories on non commutative spaces in which the ordinary product among fields is substituted by a non commutative $\star$-product

The fuzzy disc

Among the applications of non commutative geometry are fuzzy spaces. The fuzzy disc is a discretization of the algebra of functions on the two dimensional disc using finite matrices which preserves the action of the rotation group.

$$f(z,\bar{z}) = \sum_{m,n=0}^{\infty} f_{mn}\bar{z}^m z^n \Leftrightarrow \hat{f} = \sum_{m,n=0}^{\infty} f_{mn}\hat{a}^{\dagger m}\hat{a}^n.$$

Where $\hat{a}$ and $\hat{a}^\dagger$ are particular matrices which satisfy:

$$\left[\hat{a},\hat{a}^\dagger\right] = aa^\dagger - a^\dagger a = \theta\hat{1}.$$

The field theory.

We will study then a quantized φ^4 scalar field theory approximating field with $N \times N$ matrices:

$$S(\varphi) = \int (\varphi \nabla^2 \varphi + \mu \varphi^2 + \frac{\lambda}{4} \varphi^4) \mathrm{d}^2 z,$$

Using the "fuzzycation" procedure this action can be approximated by:

$$\hat{S}_N(\hat{\varphi}) = \pi \,\mathrm{Tr} \left\{ 4\hat{\varphi} \left[\hat{a}, \left[\hat{\varphi}, \hat{a}^\dagger \right] \right] + \theta \left(\mu \hat{\varphi}^2 + \frac{\lambda}{4} \hat{\varphi}^4 \right) \right\} \qquad (2)$$

In this object φ are finite dimensional matrices and the products between the field became the standard matrix multiplication, being a finite matrix model can be approached numerically using Monte Carlo techniques a method currently in use for other fuzzy spaces like the sphere.

The Monte Carlo approach

Using Monte Carlo methods we will produce a sequence of configurations and evaluate the average of the observables over the set. This Monte Carlo chains are representatives of the configuration space at given parameters. In this framework the expectation value is approximated by:

$$\langle O \rangle \approx \frac{1}{T_{MC}} \sum_{j=1}^{T_{MC}} O_j$$

For each set of fixed parameters μ, λ and N we need to run a complete independent Monte Carlo, in order to obtain just one result. The final purpose of this simulation is to explore the theory varying the parameters as much possible.

Stimulations before the GRID

The high grade of parallelism allow us to process several different configurations of the model potentially completely in parallel. Beside each Monte Carlo can be parallelized furthermore.
The first result were obtained using local resource using OpenMP/OpenMPI whit the following results:

- Computation time: 1 week
- Number of parameters sets used: 600
- Maximum matrix rank achieved: 20

Stimulations on the GRID

I was introduced to the GRID by Silvio Pardi through the Champion program. This first implementation of the GRID paradigm, using the ReCaS infrastructure, in this field of theoretical research allowed us to push the complexity of the simulations (aka the matrix rank) in such a way to obtain a much better result and even new results in the same amount of time spent in the previous simulations:

- Computation time 1 week
- Number of parameters sets used: 1200
- Maximum matrix rank achieved: 40
- total Number of jobs : 5000

Stimulations on the GRID

All the submissions were achieved using a parametric job together a suitable wrapper script which allows to use multiple parameters:

```
JobType = "Parametric";
Executable = "gridfuzzy.sh";
StdOutput = "gridfuzzy_out_PARAM_.txt";
StdError = "gridfuzzy_err_PARAM_.txt";
Parameters = 500 ;
ParameterStart = 1;
ParameterStep = 1;
Arguments = "gridfuzzy arguments.txt _PARAM_";
InputSandbox = {"gridfuzzy.sh", "gridfuzzy", "arguments.txt"};
OutputSandbox ={"out_PARAM_.tgz","gridfuzzy_out_PARAM_.txt"};
RetryCount = 10;
```

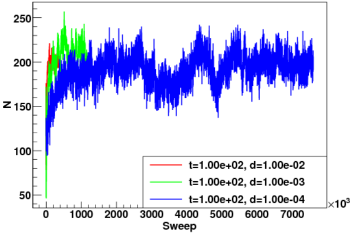
```
#gridfuzzy.sh

#!/bin/bash
exec_file=$1
arg_file=$2
N=$3
arguments=$(head -n $N $arg_file | tail -n 1)
chmod +x $exec_file #sets executable
tar -czvf out$3.tgz  *.out
rm -f *.out
exit $?
```

This first implementation we use only the parameters based parallelization obtaining excellent results although this models offer different levels of parallelism:

- Parameters based parallelism
- Mote Carlo based parallelism
- Matrix based parallelism

Mote Carlo based parallelism



After a first phase called thermalization which is necessarily serial a single Monte Carlo can be spitted in various parallel chains.

A "father“ job, after the thermalization phase, should be able to start it self new jobs using as inputs the its results and than gather the results among them. (This can be achieved using DAG jobs)

Matrix based parallelism

Another great step forward will the implementation of the matrices parallelism using the matrix $\times$ matrix parallel algorithm Will be very useful to access to resource like GPU which are very efficient for such tasks. This optimization will push the maximum matrix rank reachable. This is very desirable from a physical point of view because N is connected to the approximation of the model and to the so called continuum limit.

Conclusions and prospective

- The use of the GRID paradigm for this model brings great advantages
- This implementation can be adapted to several others simulations on high energy theory
- This implementation can be furthermore optimized