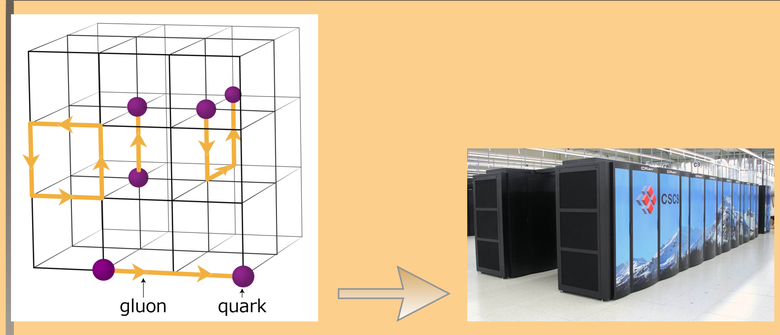
# Platform independent profiling of a QCD code

Marina.Marinkovic@cern.ch and Luka.Stanisic@inria.fr

## HPC applications in QCD




The simulations in lattice QCD:

- are large scale and require massive parallelization,
- consume a lot of computing resources,
- and require constant algorithmic development and optimization for new hardware.

## openQCD code

**openQCD** simulation package consists of several programs for generating gauge configurations with $O(a)$ *improved Wilson fermion action*, allowing for several different choices of boundary conditions:

- open,
- periodic,
- Schrödinger Functional (SF), and
- open-SF.

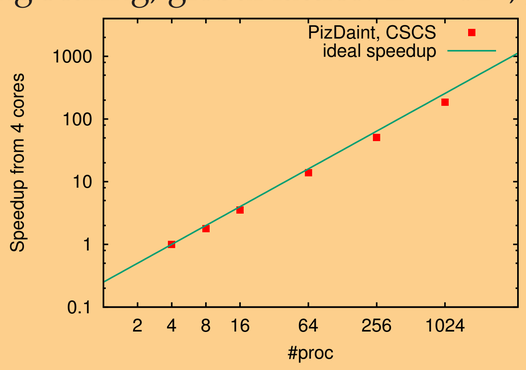
This simulation program is based on the Hybrid Monte Carlo algorithm [3] and supports parallelization in 0,1,2,3 or 4 dimensions. All the programs in this package are highly optimized for machines with current Intel or AMD processors, but will run correctly on any system that complies with the ISO C89 and the MPI 1.2 standards. The code is open source GPL, and is available for download from `http://luscher.web.cern.ch/luscher/openQCD/`

## Benchmarking: ym1

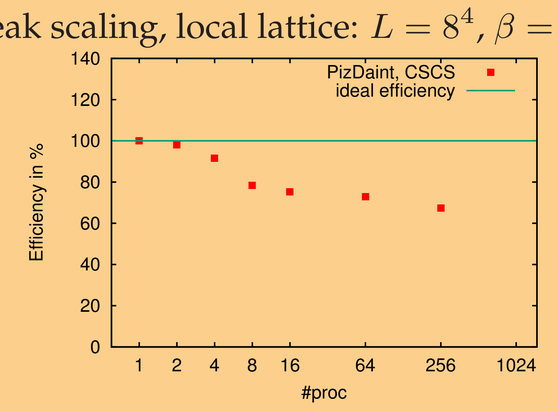
We apply the first set of benchmarking on the program `ym1` that generates an ensemble of gauge fields representative of the (pure) SU(3) gauge theory. Exactly which theory is simulated depends on the parameters passed to the program. For the tests in this presentation, we chose tree-level improved Symanzik gauge action.

## Scaling of openQCD

- Strong scaling, global lattice: $L = 32^4$, $\beta = 6.0$




- Weak scaling, local lattice: $L = 8^4$, $\beta = 6.0$




**Strong and weak scaling of the code on a typical Cray machine (PizDaint, CSCS).**

## Abstract

The supercomputing platforms available for high performance computing based research evolve at a great rate. However, this rapid development of novel technologies requires adaptations and optimizations of the existing codes for each new machine architecture. In such context, minimizing time of efficiently porting the code on a new platform is of crucial importance. A possible solution is to use coarse grain simulations of the application that can assist in detecting performance bottlenecks. We present a procedure of implementing the intermediate profiling for **openQCD** code[1] that will enable the global reduction of the cost of profiling and optimizing this code commonly used in the lattice QCD community. Our approach is based on well-known **SimGrid** simulator[2], which allows for fast and accurate performance predictions of the codes on HPC architectures. Additionally, accurate estimations of the program behavior on some future machines, not yet accessible to us, are anticipated.

## HPC Architectures

**Ever-changing HPC architectures (TOP500)**

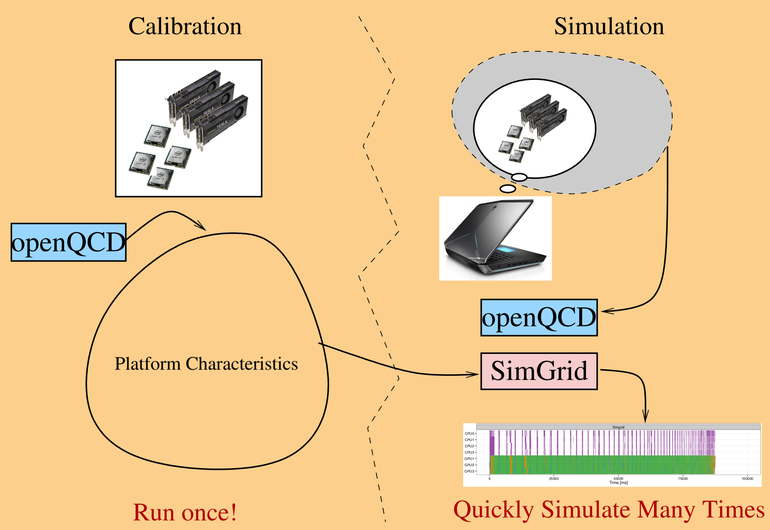
| Date | Name | CPU | Accelerator | Interconnect |
|---|---|---|---|---|
| 6/2016 | TaihuLight | SW26010 RISC | ? | Sunway |
| 6/2013 | Tianhe-2 | Intel Xeon | Xeon Phi | TH Exp.2 |
| 11/2012 | Titan | AMD Opteron | Kepler | Cray Gemini |
| 6/2012 | Sequoia | IBM Power BQC | - | Custom |
| 11/2011 | K computer | Fujitsu SPARC64 | - | Tofu |

## SimGrid Simulator

**SimGrid** is a simulation framework for distributed systems and as such, ideal for application like **openQCD** package that rely solely on MPI communication (no shared memory is assumed). Additional benefits of using **SimGrid** for the profiling of **openQCD** code:

- **SimGrid** is performance oriented and scalable.
- It has flexible flow-based contention model for communications.
- It is portable, open source and easily accessible.

## Envisioned Workflow




## Profiling with SimGrid

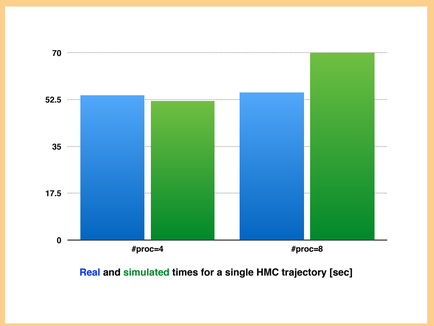
In order to develop a code which scales well on diverse platforms, besides the access to different machines, the developer needs to be able to perform large experimental campaigns in a fairly short period of time. The main advantages of using a **SimGrid** simulator are:

1. Rapid and reproducible executions.
2. Use one node instead of the whole supercomputer.
3. Predictions on unavailable architectures.

## Initial evaluation of openQCD

We tested `ym1` program in the case of 4 and 8 processors on the computing platform:

- **2 Dodeca-core Haswell Intel Xeon E5-2680**




## Next steps

- Better communication modeling
- Increase number of processors
- Calibrate other supercomputers (Stampede, Blue Waters, MareNostrum, CSCS, etc.)

The goal is to reproduce the correct scaling on a single machine and device a general method for the prediction of the scaling on an arbitrary machine.

## Future machines

The same procedure can be applied on machines that are still in the design phase, once the following features are known:

- machine topology
- processor speed
- communication characteristics

Nevertheless, it is hard to correctly model an unknown machine and the results must be interpreted carefully[4].

## Acknowledgements

This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID s642. Experiments presented in this paper were carried out using the PlaFRIM experimental testbed, being developed under the Inria PlaFRIM development action with support from LABRI and IMB and other entities: Conseil Régional d'Aquitaine, Université de Bordeaux and CNRS.

## References

[1] Luscher, Schaefer, `http://luscher.web.cern.ch/luscher/openQCD/`

[2] Casanova et al., `http://hal.inria.fr/hal-01017319`

[3] Duane, Kennedy, Pendleton, Roweth, Physics Letters B 195 (2): 216-222.

[4] Luka Stanisic, PhD Thesis. University Grenoble Alpes, 2015 `https://hal.inria.fr/tel-01248109v2`