Computational Materials Science at the Exascale

Markus Eisenbach, Oak Ridge National Laboratory

Introduction

Computer architectures are currently undergoing a paradigm shift that has been forced by hardware limitations in achieving higher computational performance. This is evidenced by the exploration of many different architectural approaches to continue the scaling of available performance. Historically similar shifts have occurred, when serial computers reached their limit and the resultant emergence and eventual dominance of distributed memory parallelism in High Performance Computing (HCP). At this time, as we explore the paths towards exascale computing, the issues that will need to be addressed mainly arise from the physical limitations imposed on the shrinking of linear dimensions of semiconductor features and the practical limitations on power consumption. Consequently applications in the next decade will need to deal with the following important issues:

- Dramatically increased parallelism
- Stagnating performance of single processing units
- Reduced amount of memory/processing unit
- Cost of data transport and need for data locality and deep memory hierarchies
- Failure of processing units during application execution

These challenges will require a fresh look at the way scientific applications are developed and a reevaluation of the algorithms we employ to solve specific scientific problems.

Scientific Need

Yet there exists a clear scientific need to go beyond the capabilities of current calculations, as can be illustrated by first principles based calculations of finite temperature magnetism in metals. Solving the problems related to the first principles understanding of real materials pose distinct challenges that go beyond today's computational capabilities. While the ground state properties of a pure compound can readily be calculated with density functional theory today, real materials have to be considered – structures with atomic impurities, crystal defects, grain boundaries and other low symmetry structures – this makes the calculation of even the ground state for realistic models for systems of >1000 atoms a daunting problem on current computer systems. The need for orders of magnitude larger computational resources arises from the need to calculate the finite temperature properties of these materials. Petaflop computers have enabled us recently to calculate the Curie temperature from first principles for simple materials such as ideal iron crystals. Realistic materials are significantly more complex than these idealized materials, consequently requiring more computational resources. A relevant length scale can be obtained by considering the thickness of a magnetic domain wall, typically a few hundred atomic layers. A computational super-cell to

describe the thermodynamics of such a system will require >100k atoms, which is 100x larger than the cell sizes for which thermodynamics is currently achievable. Going beyond the static behavior of magnetic systems requires the inclusion of magnetic kinetics in the calculations, driving the computational resources required for realistic systems at finite temperatures well into the exascale regime.

Future development needs

A major constraint for the development of many Materials Sciences research codes are limited developer resources. This will require economical approaches to address the issues lined out above that ideally will be transferable to a number of possible future architectures and allows as much code reuse from legacy codes as possible, especially in non performance critical regions.

Any new technology that will have chances of success must fulfill a number of requirements to be widely adopted and ensure its survival:

- Ease of use by application developers, who are usually domain scientists with little formal CS training.
- Portability across a wide range of platforms, vendors and scales from laptop computers to HPC.
- The perception that the technology will survive and will be widely supported.

The success of libraries such as BLAS, LAPACK and fftw in achieving portable performance should encourage the search of higher level algorithms that could benefit from supporting libraries that can provide performant implementations of commonly needed tasks without reinventing the wheel and hence free up application developer resources. An example where high level physics can be provided is the libXC library for exchange-correlation functionals for density functional theory. But not all functionality can be provided by libraries and it is important that programming models for can be integrated with existing codes and are interoperable with libraries. (E.g. use of accelerator device based BLAS with OpenACC) The most successful programming models for accelerator devices to date (CUDA and OpenCL) fall short in that they require a separate code base written in a language that is subtly different from the language that it resembles (i.e. C) and that does not provide the expressiveness that users have to come to expect from modern incarnations of languages such as C++11 or Fortran2008. Here substantial research and development is still needed to device a portable, expressive programming model that allows the organic integration of all parts of a scientific application and that can serve as a starting point for performance optimizations, such as explicit management of data transfer within the memory hierarchy or special algorithms tailored to hardware architectures. In other words the ideal model would allow the targeted low level refinement of implementations that can otherwise written in a sufficiently high level abstraction to shield large parts of the code base from general obsolescence in the case of dramatic architectural changes.