ISSN 2186-7437

NII Shonan Meeting Report

No. 2015-18

Putting Heterogeneous High-Performance Computing at the Fingertips of Domain Experts

Wim Vanderbauwhede
Sven-Bodo Scholz
Tetsuya Takemi

November 17–20, 2015

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Description of the meeting

Perspectives, needs and ideas from the research communities involved in using, operating and developing HPC software and systems.

Background

High-performance computing (HPC) is critically important in many scientific fields that use numerical models which are computationally intensive both in terms of the speed of computation and in the memory usage. For example, recently large-eddy simulation models have become more and more common for research in meteorological and atmospheric sciences because they can explicitly represent the turbulent nature of atmospheric flow and dispersion at fine spatial scales. This is a very promising tool for numerical weather forecasting, emergency response, hazard/disaster assessment, and air-quality assessment, but the technique is much more computationally expensive than the current numerical weather forecasting. Writing code that will run with good performance on modern computer systems is becoming increasingly hard because of the advent of manycore systems and an increasingly broad range of accelerators such as Graphics Processing Units (GPUs) and Field- Programmable Gate Arrays (FPGAs). HPC clusters built from such systems pose an even greater challenge. Traditionally, such highly parallel HPC systems were predominantly used by a rather small group of people. Applications were limited to a relatively well defined set of scientific codes which were tuned and adjusted over long periods of time. The most demanding applications typically were highly economically relevant and, therefore, could afford HPC experts with extensive expertise in programming parallel systems to devote large amounts of time to fine-tuning the codes to the executing hardware. Since the turn of the century, the situation has changed dramatically. Not only has the information area that we are embarking into led to new increased data processing demands, sometimes referred to as big data, but we also see heterogeneous HPC systems becoming mainstream facilities. As these systems become more affordable they become more widely available. We see a much larger cohort of application programmers from domains such as climate modelling, image processing, finance, personalised medicine, biology, chemistry, etc. The compilers and analysis tools required to achieve high performance on modern systems are highly complex in themselves, and aimed at the HPC experts rather than the domain experts. However, many domain experts do not have the means to employ HPC experts, nor can they afford the time to acquire the necessary expertise themselves. As a result, the gap between the performance of code written by many domain experts and the capability of modern computer systems is growing steadily. Even for the HPC experts it is becoming increasingly difficult to achieve optimal performance, due to the huge complexity of heterogeneous manycore systems. This is already the case for systems with accelerators such as GPUs or Intel's Many Integrated Cores (MIC) architecture, but is particularly acute for FPGAs: these devices are very promising for HPC because they can achieve very high performance per Watt; however it is still very difficult to achieve good performance in FPGA computing.

To safeguard the progress of the domain experts and, with it, the progress of scientific research relying on numerical computations, addressing this perfor-

mance gap is crucial.

Bridging the gap

The current work flow for scientific computing is typically as follows: domain experts typically write single-threaded code, usually in Fortran or C/C++. If they have in their team parallel programming expertise, or the means to afford support from HPC experts, the code will be parallelised for clusters through MPI and for multicore processors via OpenMP. Porting the code to GPUs requires manual rewriting of parts of the code in CUDA or OpenCL. For FPGAs, the situation is even more complex: here, the code needs to be re-implemented in a hardware design language. At the same time, there are many efforts in the computing science community to create languages and compilation approaches that can target heterogeneous systems without manual rewrites, for example OpenACC, Single-Assignment C, Halide, as well as languages with explicit parallism support such as Chapel, Cilk, Co-array Fortran and many others. However, this research assumes knowledge about programming paradigms, architectures, cost models etc. which come natural to computing scientists and HPC programmers but not to domain experts. We therefore want to bring these communities together to exchange views, so that the computing science research will benefit the domain experts much more directly.

Aim of the meeting

The aim of this Shonan meeting is to bring together researchers from the disciplines involved, in particular

- domain experts such as geophysicists, meteorologists etc,
- High-Performance Computing experts,
- computing scientists with expertise in programming languages and compilers for heterogeneous manycore systems,
- specifically FPGA and GPU experts

to have a discussion on the challenges each community faces and on ways to bridge the gap between the domain experts and today's and tomorrow's clusters of heterogeneous many-core systems. We want to address questions relating performance to code analysis, refactoring, compilation and run-time adaptation, as well as user experience design. The different communities will ask very different questions: computing scientists might ask: "Is static code analysis possible? Do we require run-time analysis? Should the compiler suggest changes to the source code for better performance? Do we need tools to predict performance on various platforms? Can we develop analysis tools that provide useful feedback to a non-expert end user?"; HPC experts would e.g. ask "What obstacles do we face when rewriting existing code to get top performance? Could the compiler give us suggestions based on the machine architecture?" and domain experts might ask "Why do I need to write code at all? Can we not use the equations as inputs? Why does the compiler not automatically parallelise my code? How do I select my numerical algorithm, why is there no tool to help me with this choice?" All these questions are interlinked, and the different communities will have very different perspectives on the issues involved. We want to stimulate this discussion so that the different communities can learn from one another and arrive at shared ideas that will put Heterogeneous High-Performance Computing at the fingertips of the domain experts.

Overview of Talks

Bridging the gap between the atmospheric scales: Atmospheric modeling by coupling NWP and CDF models

Prof. Tetsuya Takemi,Kyoto University

The atmospheric motions has a wide range in their temporal and spatial scales. With the continuing advances in computational resources, numerical weather prediction (NWP) models can resolve the scales on the order of 1-10 km, while computational fluid dynamics (CFD) models can cover the spatial scales on the order of 1-10 km. Therefore, there is an overlap in the scales that can be represented in both NWP and CFD models, which enhances the collaboration between atmospheric scientists and fluid engineers. In this talk, we will present our recent efforts in simulating microscale atmospheric flows over complex topography including urban districts by coupling NWP and CFD models. Specifically, turbulent flow and dispersion over complex topography was simulated with a large-eddy simulation (LES) technique in the CFD model. A special care was made to represent turbulent motions in connecting the NWP model outputs into an LES model.

Optically reconfigurable gate array for heterogeneous high-performance computing

Prof. Minoru Watanabe, Shizuoka University

Optically reconfigurable gate array (ORGA) can support a high-speed dynamic reconfiguration. The reconfiguration time of ORGAs reaches over 100 MHz. Its programmable gate array can dynamically be reconfigured at every 10 ns. Using such high-speed dynamically reconfigurable devices, single-instruction-set-computer (SISC) can be implemented. The SISC can work on dynamically reconfigurable device just like RISCs while such SISC can increase the performance of programmable gate arrays due to its simple architecture. In the first presentation, I will introduce the SISC implementation on ORGA which maybe can be applied for heterogeneous high-performance computing.

Multiparty Session Types and their applications to HPC

Prof. Nobuko Yoshida, Imperial College London

We give a summary of our recent research developments on multiparty session types for verifying distributed and concurrent programs, and our collaborations with industry partners. We shall first talk how Robin Milner, Kohei Honda and Yoshida started collaborations with industry to develop a web service protocol description language called Scribble and discovered the theory of multiparty session types through the collaborations. We then talk about the recent developments in Scribble (which is a protocol description language for multiparty session types), the runtime session monitoring framework and applications to generate MPI programs from Scribble.

Climbing Mont Blanc - A Training Site for Energy Efficient Programming on Heterogeneous Multicore Processors

Prof. Lasse Natvig,Norwegian University of Science and Technology

Climbing Mont Blanc (CMB) is an open online judge used for training in energy efficient programming of state-of-the-art heterogeneous multicores. It uses an Odroid-XU3 board with an Exynos Octa processor and integrated power sensors. This processor is three-way heterogeneous containing 14 different cores of three different types. The board currently accepts C and C++ programs, with support for OpenCL v1.1, OpenMP 4.0 and Pthreads. Programs submitted using the graphical user interface are evaluated with respect to time and energy used, and energy-efficiency (EDP). A small and varied set of problems are available, and the system is currently in use in a medium sized course on parallel computing at NTNU. Other online programming judges exist, but we are not aware of any similar system that also reports energy-efficiency. The talk will present some early experience from using the CMB system and explain how fellow researchers can collaborate and contribute by uploading new problems and solve existing problems. Our long term goal is to enhance the body of knowledge in the area of energy-efficient computing on handheld devices from submissions to the system.

Towards a HPC Research Roadmap Beyond Exascale

Prof. Dr. Theo Ungerer,University of Augsburg

The goal of EuroLab-4-HPC, a European Community funded CSA (Coordination and Support Action), is to create the foundation for a European Research Center of Excellence in High Performance Computing Systems. One of the principal objectives of the project is to align the agendas of the best research groups on a roadmap for long-term HPC system research. The HPC research roadmap of EuroLab-4-HPC targets a long-term research vision beyond exascale (2020 to 2030). This roadmap will include all layers of the HPC stack, from applications to hardware, as well as the vertical challenges of Green ICT, energy and resiliency, and the convergence of HPC, embedded HPC, and data centres for big data. We are currently forming cross-cutting working groups to address the problem from these diverse perspectives. A roadmap that targets five to fifteen years in the future will naturally contain parts that are highly speculative. We are therefore identifying disruptive technologies that could be technologically feasible within the next decade, in order to assess how they would affect future architectures and the research roadmap. The talk will focus on disruptive technologies and will discuss our initial directions with the audience. The roadmapping efforts started Sept 2015; a preliminary roadmap is due August 2016 and the final roadmap will be delivered on August 2017. The HPC research roadmap is coordinated by Theo Ungerer, University of Augsburg.

Safe embedding and optimizing of EDSL

Prof. Oleg Kiselyov,Tohoku University

We outline a framework for embedding of typed domain-specific languages that ensures that only well-typed DSL expressions are embeddable. The framework lets the user write a wide range of transformations. They typically convert a DSL expression to a form from which an efficient target code can easily be generated. The transformations preserve typing and hygiene by the very construction, and hence make it more difficult for the transformation writers to shoot themselves in the foot. We have used the framework for compiling language-integrated queries to efficient SQL.

Performance-portable HPC code from single source code

Dr. Mark Govett, NOAA

I would like to talk about our experience developing code that is performance portable with a single source code, and runs efficiently on CPU, GPU and MIC systems. It has been run on over 130,000 CPU cores, 15000 GPUs, and over 1000 MIC processors. I will show scalabiliy and performance results for our Weather model. But I am most interested in learning and having a conversation where we share our collective knowledge on techniques, code design, algorithms, etc that are necessary elements toward achieving performance portability with a single source.

Practical abstract machine models for heterogeneous multicores

Dr. Raphael 'kena' Poss, University of Amsterdam

It is 2015 and yet we are still subject to a 1980-era diversity of frameworks and programming languages each with their own abstractions for programmers. One supports GPU-like parallelism, another can offload certain streaming operations (eg. crypto), another offers on-chip message passing... But what is the underlying structure behind? In this talk I will highlight commonalities and differences and present our current research direction with AM3, a general-purpose model for heterogeneous platforms.

Building blocks for domain experts: a compiler and runtime system perspective

Prof. Albert Cohen, INRIA

Performance portability is always important, but the needs of domain experts go beyond achieving decent performance at a reasonable cost. Our research aims at helping tool designers rather than providing multi-purpose or specialized tools for domain experts themselves. In particular, we are interested in improving the productivity of programming language engineers who are also interested in a specific domain, and domain engineers interested in developing programming language tools. Eventually, ninja domain programmers being the most influential in defining the software practices in high-performance computing, they tend to also determine the success or demise of a computing platform. We are thus helping computing platform engineers increase their ability to provide great tools for their ninja users. We will take concrete examples from polyhedral compilation, task-parallel runtimes, and cyberphysical systems with a strong computational component, raising open questions towards the adoption of such techniques and looking for collaborations.

High level programming in heterogeneous cluster environments

Dr. Oren Segal, University of Massachusetts Lowell

The strength of heterogeneous systems is also their Achilles heel, i.e. the diversity of the devices and ecosystems needed to maintain them present major technological challenges. Some of the biggest challenges are in the realm of system programing. We believe that for heterogeneous systems to become a mainstream design choice, high level and standard software design flows need to be adopted in order to achieve transparency when dealing with diverse devices and architectures. We present two open source frameworks meant to assist in transparency when dealing with accelerators in common data center environments. The first is Aparapi-UCores which allows automatic Java to OpenCL translation and targeting of CPUs/GPUs/APUs and FPGAs using a single Java code base. The second is Spark-UCores (SparkCL) which allows running Java/OpenCL accelerator code on heterogeneous cluster nodes running as part of an Apache Spark cluster. We describe the current status of these frameworks and performance results across different architectures. We will also discuss, caching, scheduling and challenges related to achieving good performance and performance per watt on heterogeneous clusters using a high level programming framework such as SparkCL.

Chapel: A Programming Language for Productive, Future-Proof Parallel Computing

Dr. Brad Chamberlain, Cray Inc

In this talk, I will describe some of Chapel's core features for productivity and flexibility, such as the ability for end-users to define their own parallel loop schedules and array distributions within the language. I'll provide status and future plans for the language and will call out ways in which we hope that applied scientists and other computer scientists might work with us to increase the chances of productive exascale computing with Chapel.

Theories and Optimization of Nonuniform Locality and Heterogeneous Memory

Prof. Chen Ding, University of Rochester

Formal definitions of locality at program, trace, and machine levels. The higher-order theory of locality (HOTL) and its use in cache performance optimization especially for multicore/manycore shared cache. (Exploring with others) Formalization and theoretical implications of locality optimization in parallel applications, languages, compilers and run-time systems. The safe parallelization of a scripting language, in particular Ruby, and a demonstration of safe (hint-based) parallel programming.

Acceleration of Global Atmospheric Model by Heterogeneous Computing with OpenACC

Prof. Ryuji Yoshida, RIKEN

1. an example of the leading edge global atmospheric simulation 2. climate simulation needs more computational performance - it's a motivation for using heterogeneous HPC. 3. an example of applying heterogeneous computing for global atmospheric model (NICAM-DC) - the model is implemented by using OpenACC, and I'll talk about a tuning strategy. 4. an another example: accelerator would be effective for physical schemes (radiation scheme) 5. current difficulties in heterogeneous computing 6. As a discussion topics; what is the most effective heterogeneous computing? - I would like to introduce a project trying to use DSL on the global model.

Homogeneous Software for Heterogeneous Architectures

Dr. Dan Ghica, University of Birmingham

The increasing complexity of hardware (e.g. GPU and FPGA) and OS-level software (e.g. containers) is reflected in a proliferation of new programming languages and idioms. Even though many such languages enjoy all the syntactic trappings of modern programming languages they are essentially system-specific languages which exhibit most of their typical problems (extreme complexity, fragility, lack of portability) and impose a tremendous burden on the programmer. It is not an exaggeration to say that many such emerging computational platforms are inaccessible to all but a few highly trained specialists. There is a clear danger that we will repeat the mistakes of the pioneering days of computing (1950s and 1960s) when each computer model came with its own distinct OS and programming language. We need to re-establish machine-independent programming as the default and dominant programming model, for the same reasons that it was first established in the 1960s with the advent of Fortran, Algol and LISP. In this talk I will explain how the programmer can be relieved of the burden of programming heterogeneous architectures by creating better type systems, more powerful compilers and smarter linkers. I will illustrate the theory with two concrete case studies: compiling an Algol-like programming language for the heterogeneous Zinq architecture from Xilinx, and an ML-like programming language for a generic deployment-independent distributed platform.

Present and future projections of urban climate based on downscaling techniques

Prof. Satoru Iizuka, Nagoya University

Present and future (2030s, 2050s, and 2070s) projections of urban climate in the third biggest metropolitan area in Japan (Nagoya metropolitan area) based on downscaling techniques are presented. Downscaling is a computer simulation technique to systematically analyze/project present or future climate from global scale to urban scale in a step-by-step manner. In this presentation, first, the prediction accuracy of the downscaling simulations is verified by comparing observation data and the results of the present climate analyses, and the advantages and disadvantages are studied. Next, the future climate change in the Nagoya metropolitan area is examined through the results of the future climate projections. In future climate projections, it is necessary to assume various future scenarios such as global climate scenario (greenhouse gas emissions scenario), urban structure scenario, and city-block/building structure scenario. The effects of the future scenarios on the future climate projections and the uncertainties of the projections are also discussed.

The Heterogeneous Programming Stack: Architectures, Performance Monitoring, Scheduling, and Virtualization

Prof. Lesley Shannon, Simon Fraser University

As we move into a heterogeneous computing world, it introduces changes to all aspects of the computing hierarchy. While we work to make changes to the individual components there is also significant potential to improve computing performance, power efficiency, and programmability by working on the interactions and integration of the various components of the computing stack. This talk discusses the some of the questions that should be addressed to frame this next generation of computing design.

Heterogeneous Computing without Heterogeneous Programming?

Dr. Clemens Grelck, University of Amsterdam

Heterogeneous computing systems offer unprecedented performance at the expense of unprecedented programming complexity. A range of programming paradigms need to be mastered and carefully integrated with each other to harness at least some of the compute power theoretically offered by such systems. This may be a challenge for some brave, but hardly more than a nuisance for the vast majority of programmers. I will talk about our work in the context of the functional array language SAC (Single Assignment C) to make heterogeneous systems effectively usable without any sort of heterogeneous programming. From a uniform source code that is highly abstract, but does expose fine-grained concurrency, we aim at compiling down to heterogeneous systems in a completely automatic way. While this may not achieve the highest possible performance, we open up a door for non-expert programmers in a world of ever more complex computing systems.

List of Participants

- Mark Govett, NOAA ESRL, USA
- Lesley Shannon, Simon Fraser University, Canda
- Satoru Iizuka, Nagoya Universit,y Japan
- Ryuji Yoshida, RIKEN Advanced Institute for Computational Science, Japan
- Nobuko Yoshida Imperial College London, UK
- Albert Cohen, INRIA, France
- Clemens Grelck, University of Amsterdam, The Netherlands
- Bradford Chamberlain, Cray Inc., USA
- Chen Ding, University of Rochester, USA
- Oleg Kiselyov, Tohoku University, Japan
- Martin Elsman, University of Copenhagen, Denmark
- Theo Ungerer, University of Augsburg, Germany
- Yukiyoshi Kameyama, University of Tsukuba, Japan
- Oren Segal, University of Massachusetts Lowell, USA
- Minoru Watanabe, Shizuoka University, Japan
- Lilia Georgieva, Heriot-Wattt University, UK
- Christian Fensch, Heriot-Watt University, UK
- Mary Sheeran, Chalmers University of Technology, Sweden
- Lasse Natvig, IDI NTNU, Norway
- Raphael 'kena' Poss, University of Amsterdam, The Netherlands
- Dan Ghica, University of Birmingham, UK
- Cosmin Oancea, University of Copenhagen, Denmark
- Hisashi Yashiro, RIKEN Advanced Institute for Computational Science, Japan

Meeting Schedule

Check-in Day: November 16 (Mon)

- Welcome Banquet

Day1: November 17 (Tue)

- Talks and Discussions
- Group Photo Shoot

Day2: November 18 (Wed)

- Talks and Discussions

Day3: November 19 (Thu)

- Talks and Discussions
- Excursion and Main Banquet

Day4: November 20 (Fri)

- Talks and Discussions
- Wrap up