Introduction to RADR 2019

Pete Beckman, Emmanuel Jeannot, Swann Perarnau

Workshop on Resource Arbitration for Dynamic Runtimes (RADR)

Pete Beckman
Argonne National Laboratory
Northwestern University
Argonne, IL, USA
beckman@anl.gov

Emmanuel Jeannot
TADaaM Team, Inria
Talence, France
emmanuel.jeannot@inria.fr

Swann Perarnau
Argonne National Laboratory
Argonne, IL, USA
swann@anl.gov

Abstract

The question of efficient dynamic allocation of compute-node resources, such as cores, by independent libraries or runtime systems can be an nightmare. Scientists writing application components have no way to efficiently specify and compose resource-hungry components. As application software stacks become deeper and the interaction of multiple runtime layers compete for resources from the operating system, it has become clear that intelligent cooperation is needed. Resources such as compute cores, in-package memory, and even electrical power must be orchestrated dynamically across application components, with the ability to query each other and respond appropriately. A more integrated solution would reduce intra-application resource competition and improve performance. Furthermore, application runtime systems could request and allocate specific hardware assets and adjust runtime tuning parameters up and down the software stack.

The goal of this workshop is to gather and share the latest scholarly research from the community working on these issues, at all levels of the HPC software stack. This include thread allocation, resource arbitration and management, containers, and so on, from runtime-system designers to compilers. We will also use panel sessions and keynote talks to discuss these issues, share visions, and present solutions.

Scope

Over the last five years, the number of nodes in large supercomputers has remained largely unchanged. In fact, the Oak Ridge National Laboratory computer leading the Top500 list, Summit, has fewer nodes than its predecessor, which is 20 times slower. Machines are getting faster not by adding nodes, but by adding parallelism, cores, and hierarchical memory to each compute node. This shift in how computers are scaled up makes it imperative that parallel computer resources within a node be carefully orchestrated to achieve maximum performance. Dynamically allocating and managing threads and the mapping of these threads to cores is a challenge that requires cooperation and coordination between the different components of the software stack.

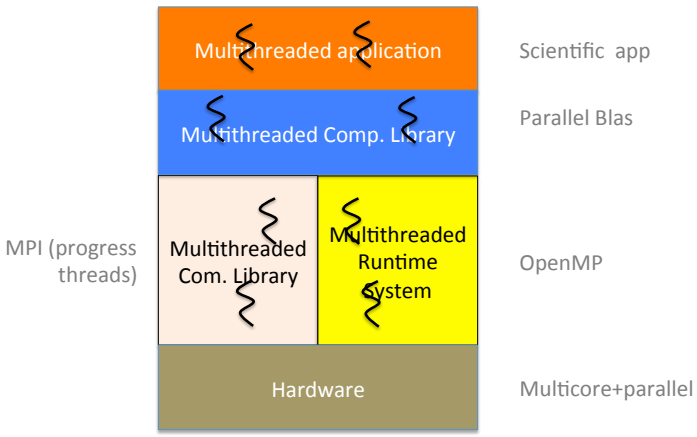

Figure 1: Software stack with different parts using threads

At the application level, a software component might use pthreads to express and coordinate concurrency. The application might also be linked to computational libraries, such as PETSc or Intel's MKL that could be multithreaded. Moreover, other parts of the application may use OpenMP parallel section (which are implemented with threads). Furthermore, the runtime system may need its own parallel resources, for example, to spawn progress engines for message libraries or remote invocation handlers. Currently, each component in this complex software stack is unaware of the other pieces. Therefore, threads can compete for cores and cause profound slowdowns to intranode collective operations and barriers. Moreover, currently, no mechanisms exist to query for unused cores, to reserve some of them, or to check which part of the application is using them. Resource allocation and partitioning by the operating system must be adaptive and well connected to user-level software components.
There has been research progress in this field. Tools such as hwloc can provide information on systems, such as topology. However, hwloc is not designed to handle direct allocation and partitioning resources. Likewise, it does not provide the interfaces required for software components to negotiate how to improve performance of the application through cooperative sharing of resources. Other approaches have been proposed, such as application composition, dynamic topology management or topology-aware core selection. Resource partitioning, enforced by the operating system using containers or within multi-kernels are also being investigated. Each of those approaches brings its own set of benefits and challenges, that need to be discussed within the community, compared with each other, and evaluated against benchmarks and use cases yet to be identified. As a relatively new and specific research area, it is difficult for researchers to find a place where to submit papers and discuss solution with the whole community. Therefore, we think it is of great interest for the HPC community to provide a venue to present these work in all their specificity and foster new discussions.

Program Committee

Program Chairs:

- Pete Beckman
- Emmanuel Jeannot

Publicity Chair:

- Swann Perarnau

Program Committee:

- Dorian Arnold, Emory University.
- Denis Barthou, Bordeaux INP.
- Siegfried Benkner, University of Vienna.
- George Bosilca, Univ Of Tennessee.
- James H Cownie, Intel.
- Carter Edwards, Nvidia.
- Hal Finkel, Argonne Ntl Lab.
- Karl Fuerlinger, LMU, München.
- Balazs Gerofi, U. Tokyo – Riken.
- Brice Goglin, Inria.
- Raymond Namyst, Univ. Of Bordeaux.
- Stephen Olivier, Sandia Ntl Lab.
- Tapasya Patki, Lawrence Livermore Ntl Lab.
- Marc Perache, CEA.
- Swann Perarnau, Argonne Ntl Lab.
- Rolf Riesen, Intel.
- Sameer Shende, U. of Oregon.
- Christian Terboven, RTW Aachen.

Program

This year, we have selected one paper to be presented at this workshop and invited three keynote speaker. The topics covers by these talks will show the importance and the variety of the RADR problematic.

We hope to see you at this first edition of RADR and tat this workshop will be a place for fruitful and lively interactions.

RADR 2019 Invited Speaker

Dynamic Resource Management: An Application Perspective

Anshu Dubey
Mathematics and Computer Science Division, Argonne National Laboratory
Flash Center for Computational Science, University of Chicago

Abstract

The state of practice among system software developers and applications developers is largely that of mutual exclusion. System software aims to make its services transparent to the applications codes, while applications tend to do everything on their own. In reality neither is optimal. Applications know where there are most fruitful opportunities for resource and data orchestration. When the applications do it themselves they may eliminate the possibility of using future developments in system software that may be beneficial. And when system software does is transparent to the applications, it leaves a lot of domain specific knowledge on the table that could have been exploited for better performance. In this presentation I will present a resource orchestration approach that we are developing for FLASH, a Multiphysics Multicomponent application code. This approach relies upon close cooperation between applications and tools developers, and aims to be as future-proof as possible given what we know of machine and software development trends.

Bio

Anshu Dubey is a computer scientist in the mathematics and computer science division at Argonne National Laboratory and a Senior Scientist at large at the University of Chicago. She leads the earth and space sciences sub-area of applications development in the US-DOE Exascale Computing Project. She is also the chief software architect for FLASH, a multiphysics multiscale HPC software that is used by multiple science and engineering domains as their community code. She is interested in all aspects of HPC scientific software including numerics, design and productivity issues.

RADR 2019 Invited Speaker

An Outlook to Node Resource Management in HPC: Challenges and Opportunities.

Balazs Gerofi
Riken

Abstract

Node architecture in high-performance computing (HPC) is becoming increasingly complex. Heterogeneous processing elements combined with deep memory hierarchies and complex NUMA topologies require careful mapping of application components to node resources so that efficient utilization of the hardware can be attained. Applications, on the other hand, are also increasingly complex and dynamic in behavior which further complicates efficient resource utilization. This talk will address multiple aspects of the node resource management issue and is comprised of three parts. Two ongoing efforts at RIKEN will be discussed as motivating case studies: mpipin, a portable, MPI implementation agnostic process and workflow placement tool, and the utility thread interface (UTI) that provides and API for applications/libraries to better describe thread behavior to the underlying operating system. Finally, a recent international effort that aims to combine multiple existing resource management pieces (including the two RIKEN ones) will be introduced.

Bio

Dr. Balazs Gerofi is a research scientist at the RIKEN Center for Computational Science, where he is involved with system software research and development for high performance computing. He actively participates in the design and development of the Post K supercomputer, Japan's next-generation flagship supercomputer after the K Computer. Balazs earned his M.Sc. degree and Ph.D. degree in computer science from the Vrije Universiteit Amsterdam and The University of Tokyo, respectively. His research interest covers operating systems, high performance computing, cloud computing, and fault-tolerant computing. Balazs is a member of the IEEE Computer Society and the Association for Computing Machinery (ACM).

RADR 2019 Invited Speaker

An Update on the Qthreads Lightweight Threading Library

Ron Brightwell
Sandia National Laboratories

Abstract

This talk will discuss recent work to integrate the Qthreads lightweight threading library into two open source MPI implementations as part of the US DOE Exascale Computing Project. I will also discuss several recent workshops focused on identifying and prioritizing resource management challenges for future extreme-scale computing systems, many of which will require dynamic adaptive lightweight runtime systems.

Bio

Ron Brightwell leads the Scalable System Software Department at Sandia National Laboratories. After joining Sandia in 1995, he was a key contributor to the high-performance interconnect software and lightweight operating system for the world's first terascale system, the Intel ASCI Red machine. He was also part of the team responsible for the high-performance interconnect and lightweight operating system for the Cray Red Storm machine, which was the prototype for Cray's successful XT product line. The impact of his interconnect research is visible in technologies available today from Bull, Intel, and Mellanox. He has also contributed to the development of the MPI-2 and MPI-3 specifications. He has authored more than 115 peer-reviewed journal, conference, and workshop publications. He is an Associate Editor for the IEEE Transactions on Parallel and Distributed Systems, has served on the technical program and organizing committees for numerous high-performance and parallel computing conferences, and is a Senior Member of the IEEE and the ACM.