SC 2013 BoF Session "Towards Exascale Runtime Systems – Challenges and Opportunities"

Short Session Report

The BoF was held November 19 between 5:30 pm and 7:00 pm in rooms 503/504. Attendance did vary, with 25 persons present at the beginning, and approx. 40 persons towards the end. The after-session discussion went on until about 7:45 pm.

The first presentation by Robert Wisniewski (Intel Corporation) titled "*The Grand Unification*" discussed the merits and risks of a revolutionary approach to solve the Exascale software problems vs. an evolutionary method, with focus on programming models and the underpinning runtime systems. Robert concluded that a blend of both approaches will be necessary, by building a framework which will continue to support the existing models and applications, facilitates incremental improvement wrt. scalability and efficiency, and at the same time allow for new and disruptive paradigms to be implemented.

Prof. Jesus Labarta of Barcelona Supercomputing Centre shared his view on "*Challenges and Opportunities*" for Exascale runtime systems. A change in culture of HPC application developers is needed: instead of understanding in detail how a target system works and then manually adapting applications to that mental model, developers have to "let loose", focus on the semantics of their algorithms and let future runtime systems take care of optimization and scheduling on a wide range of target systems.

Prof. Raymond Namyst of INRIA Bordeaux focused on the "*Challenges and Opportunities*" originating from established trends in system architecture, including extreme parallelism, need to minimize data movements and to integrate visualization and I/O. Programming models and runtime system stacks have to ensure effective and efficient composability between application components and parallel libraries written in different paradigms. The OCR initiative will solve this problem (in a somewhat revolutionary way), yet there is also value in agreeing on a common "micro-runtime" specifically addressing the composability problem within the plethora of current RTS solutions.

Prof. Yonghong Yan of the University of Houston derived the "*Challenges and Opportunities*" from the evolution of programming models in HPC and Big Data, and from the emergence of heterogeneous, many-core systems. He developed a food metaphor for (HPC) programming, and highlighted the need for new node-level runtime systems that fully support high-level programing models, decoupling the application from specific system details.

Prof. Toshio Endo of Tokyo Tech gave a survey of the local work on runtime systems, including the management of increasingly complex memory hierarchies, multi-level check pointing and scalable I/O architecture.

Rob Knauerhase from Intel Labs did highlight the main challenges to derive from the need for power-efficient computing, which will cause a large increase in concurrency with potentially decreasing per-core performance. These machines will require a fine-grained, event-driven execution model (rather than the synchronized BSP-style model used today), such as being implemented in the Open Community Runtime initiative. He sketched current research in the DoD ACR project on exploring dynamic tradeoffs between performance, power and reliability, and closed with a list of challenges, which amongst others did include the need to support irregular workloads and to cope with changing system policies, like adapting to changes in available power.

Prof. Guang Gao of the University of Delaware cited an example from Chinese history – many attempts at defending against floods of China's great rivers did fail, until it was realized that diverting the waters into irrigation systems would prevent floods and utilize the regular snowmelts for something good. Applied to HPC and Big Data, this means turning the challenges (such as extreme parallelism, heterogeneity and variability of systems) into assets by innovative system architecture, runtime and programming models.

The discussion after the presentations focused around two topics: how much compiler work is needed to enable "ordinary" SW developers to write highly concurrent applications, and how can the community ensure that novel approaches like OCR are applicable to a wide range of system and CPU architectures, rather than being handcrafted for specific systems.