



Qthreads: Run Time Library Support for Task Parallel Programming

Ron Brightwell and Stephen Olivier
Scalable System Software Department
Center for Computing Research

CEA/NNSA
June 2016



*Exceptional
service
in the
national
interest*



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

What Qthreads Is

- Qthreads is a user-level library for lightweight massive on-node multithreading, accessible directly using a C API or indirectly through a variety of higher-level programming models
- Initial motivation was to support graph processing / analytics applications on commodity hardware using runtime-managed massive multithreading and rich synchronization primitives
- Evolved into a dual-use software capability for both analytics and computational science

What Qthreads Is Not

- A mechanism for taking over the world
- A PR vehicle for convincing program managers that we're doing something fundamentally new in lightweight threading

Many Other Run Time Systems

- Nanos++, StarSS, OmpSS (BSC)
- StarPU, ForestGOMP (Inria)
- Habanero/OCR (Rice U.)
- SWARM/OCR (ETI)
- MassiveThreads (U. Tokyo)
- Cilk & Cilk Plus (MIT & Intel)
- MPC (CEA)
- Grappa (UW/PNNL)
- Argobots (ANL)
- Charm++ (UIUC)
- HPX (LSU & IU)
- Legion (Stanford)
- Uintah (U. Utah)

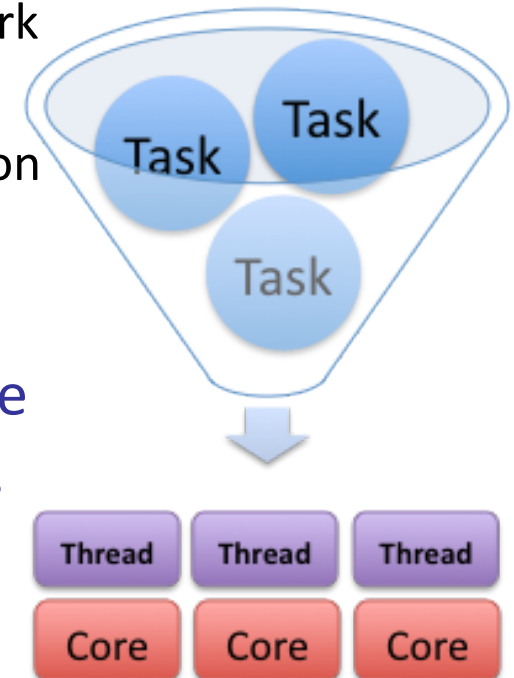
Qthreads Objectives and Outcomes



- Be efficient for both analytics and computational science
 - Many other run time systems address only one or the other
- Be a vehicle for run time system research
 - Enable **co-design** leveraging Sandia expertise across the system stack
 - Use modularity for flexibility, extensibility
 - Support for diverse architectures and programming models
- Improve understanding of system and application behavior
 - Test and challenge efficiency and scalability limits
- Impact deployed run time system technologies
 - Develop solutions to unsolved problems in adaptive run time systems
 - Present **lessons learned** to industry and the community
 - Vendors apply our techniques to their implementations, e.g., in OpenMP
 - In the case of the Chapel language, Cray has adopted our run time

Qthreads Model

- The programmer exposes application parallelism as massive numbers of lightweight tasks (qthreads).
 - Problem-centric, rather than processor-centric work decomposition, for transparent scaling
 - **Full/empty bit primitives** for flexible synchronization
 - Emulates Cray XMT (ThreadStorm) architecture
 - C API with **no special compiler support required**
- The run time system dynamically manages the scheduling of tasks for locality and scalability.
 - Persistent pthreads execute the user's tasks
 - Pinned to cores in topology-aware configuration
 - **Lightweight task switching**



Qthreads-based Software

- Chapel programming language
 - Chapel is Cray's next generation parallel programming language.
 - Qthreads provides the default tasking layer for the Chapel run time.
- Multithreaded graph library (MTGL)
 - MTGL is Sandia's toolkit for graph processing algorithms.
 - Qthreads allows MTGL, originally designed for the Cray XMT, to execute on commodity systems such as x86 and POWER machines.
- Kokkos task-parallel extensions *[In progress through ATDM]*
 - Kokkos is Sandia's C++ library for efficient management of data layout and parallelism for manycore processors.
 - Qthreads supports extensions to Kokkos that add task parallelism (futures, dependences) to existing Kokkos data parallel capabilities.
- Qthreads also served as an OpenMP run time using the LLNL ROSE compiler as the front-end. *[Not currently funded work]*

Lightweight Threading Issues

- Interaction with network [CSSE]
 - Support for lightweight thread sleep/wakeup mechanisms
 - Eliminate polling for completion and data movement
 - Strategies for mitigating lack of parallelism in network interface
 - Better network interface support for lightweight threads
- Interaction with OS [FY17 ATDM]
 - Support for introspection
 - Event-driven behavior
 - Dealing with blocking system calls
- Tools and tool infrastructure
 - Performance analysis
 - Correctness analysis

ISC Workshop on Dynamic Runtimes



Session Details



Name:  Building a European/American Community for the Development of Dynamic Runtimes in Extreme-Scale Systems

Time:  Thursday, June 23, 2016
02:00 pm - 06:00 pm



Room: Basalt
Frankfurt Marriott Hotel



Breaks: 04:00 pm - 04:30 pm Coffee Break

Organizer: Siegfried Benkner, University of Vienna
Robert Clay, Sandia National Laboratories
Bill Feiereisen, Modelado.Org







Abstract: Three main challenges face the high performance computing community on the road to exascale: 1) Exploitation of extreme parallelism and the concurrent difficulty of programming, 2) Regular hardware and software failures of these very high part-count machines requiring adaption to frequent failures incidents, and 3) Severe limits on power consumption in order to fit within datacenters' capacity and budgets. Dynamic runtimes have emerged as tools to manage quickly changing circumstances. Several European and American projects have developed prototype systems that share some common characteristics, but there is no overall consensus on the exact needs and requirements of systems that will eventually be implemented in the extreme scale systems of the next decade. This workshop will bring together the developers of several systems to discuss the building of a collaborative community. In this workshop the developers of the OmpSs programming model from the Barcelona Supercomputing Center, the StarPU system from INRIA, the Open Community Runtime from Rice University and University of Vienna and researchers from the American Department of Energy projects including HPX, DARMA and Legion will discuss these challenges and identify common threads upon which to build future collaboration across the Atlantic.

Targeted Audience

This workshop targets runtime developers in both Europe and the US who are interested in sharing common experiences, and who wish to work together toward a common set of goals. All interested runtime and applications workers are very welcome and encouraged to attend and join this community.

For more details, please visit the workshop webpage at
<http://www.modelado.org/uncategorized/building-a-europeanamerican-community-for-the-development-of-dynamic-runtimes-in-extreme-scale-systems/>

Available Online



<http://www.github.com/Qthreads>