The Uintah Framework: A Unified Heterogeneous Task Scheduling and Runtime System for Energy Applications

Alan Humphrey, Qingyu Meng and Martin Berzins. Scientific Computing and Imaging Institute, University of Utah

Introduction and Motivation

Traditional HPC systems now commonly augmented with graphics processing units (GPUs), and soon other co-processor designs such as Intel's Many Integrated Core (MIC) architecture

DOE Titan (ORNL)
- 18,688 Cray XK6 compute nodes
- 299,008 AMD Interlagos CPU Cores
- 10,000+ Nvidia Tesla K20 GPUs

Estimated Peak Performance:
- 20 Petaflops

TACC Stampede
- 6,400 Dell DCS Zeus compute nodes
- 102,400 Intel Sandy Bridge CPU Cores
- + Intel Many Integrated Core (MIC)
- 300-400k additional x86 cores

Estimated Peak Performance:
- 2 Petaflops from CPU cores
- 8 additional Petaflops w/ co-processors

Exascale Energy Problem Design Alstom Clean Coal Boiler

- Need LES resolution for 350 MW problem
- 9 x 10^{12} simulation cells
- Estimated 50-100 million cores to simulate problem in 48 hours of wall clock time

•As the ongoing convergence between multi-core CPUs, GPUs and other co-processors designs continues, it is imperative to make use of hybrid architectures.

•Task-based codes like Uintah are very well placed to exploit such architectures.

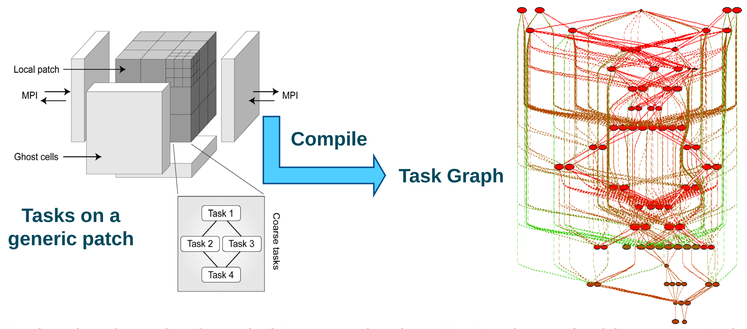
Problem and Challenges

Difficult to design general purpose software frameworks for emerging and future heterogeneous systems at multi-petaflop and eventually exaflop scales.

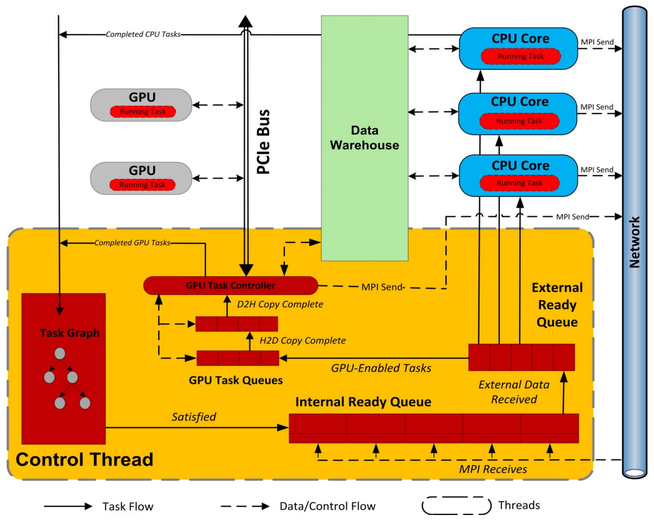
- Must address formidable scalability challenges
- Manage MPI communication and device data transfers
- Overlap computation, communication & PCIe transfers
- Shield application developer from underlying parallelism
- Mixture of concurrency APIs (MPI, Pthreads, CUDA)
- Multiple levels of parallelism
- Must effectively utilize all computational resources
  - CPU cores and GPUs simultaneously
- Fully automated load balancing

Solution: A Unified Approach

Uintah can simulate large-scale complex science-based energy systems on up to 262K cores on the DOE Jaguar system by using a novel asynchronous task-based approach



- Task – basic unit of work (C++ method or CUDA kernel with computation)
- Uintah can be generalized to support accelerator & co-processor designs
- Our hybrid runtime system and multi-threaded MPI task scheduler provides a **unified approach to heterogeneous processing** for Uintah, overlapping communication, computation & PCIe transfers
- Uintah can now fully exploit heterogeneous architectures with automated support for asynchronous, out-of-order scheduling of both CPU and GPU computational tasks.



Unified Task Scheduler and Runtime System

Supported By:

• This work was supported by DOE INCITE award CMB015 for time on Jaguar and DOE NETL for funding under NET DE-EE0004449.
• This research also used resources from the TitanDev program at Oak Ridge National Laboratory (ORNL), under Directors Discretionary Allocation, CMB021. as well as resources of the Keeneland Computing Facility at the Georgia Institute of Technology, which is supported by the National Science Foundation under Contract OCI-0910735.
• Continuing work will use DOE ALCC allocation CMB026 for time on Titan when it becomes available in late 2012

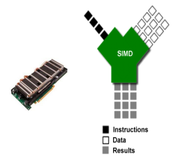
A Radiation Transport Model

For a portion of the Alstom Boiler Simulation, it is necessary to approximate the **Radiative Transfer Equation** within the ARCHES combustion component

$$\frac{1}{c}\frac{\partial}{\partial t}I_v + \hat{\Omega}\cdot\nabla I_v + (k_{v,s}+k_{v,a})I_v = j_v + \frac{1}{4\pi c}k_{v,s}\int_{\Omega} I_v d\Omega$$

In [1], our initial hybrid runtime design was examined in the context of a developing, hierarchical GPU-based ray tracing radiation transport model. This GPU-based Reverse Monte Carlo Ray Tracer is computationally intensive and ideal for SIMD parallelization

- Rays mutually exclusive
- Can be traced simultaneously
- Offload ray tracing to GPUs
- CPU cores do other computation



Results

Our scheduler design was tested by running on a single compute node and at full scale on the modern heterogeneous systems, Keeneland and TitanDev. On these systems, we demonstrated significant speedups per GPU against a standard CPU core

Single CPU Core vs Single GPU

Machine	Rays	CPU (sec)	GPU (sec)	Speedup (x)
Keeneland 1-core Intel	25	28.32	1.16	24.41
	50	56.22	1.86	30.23
	100	112.73	3.16	35.67
TitanDev 1-core AMD	25	57.82	1.00	57.82
	50	116.71	1.66	70.31
	100	230.63	3.00	76.88

All CPU Cores vs Single GPU

Machine	Rays	CPU (sec)	GPU (sec)	Speedup (x)
Keeneland 12-cores Intel	25	4.89	1.16	4.22
	50	9.08	1.86	4.88
	100	18.56	3.16	5.87
TitanDev 16-cores AMD	25	6.67	1.00	6.67
	50	13.98	1.66	8.42
	100	25.63	3.00	8.54

- **GPU – Nvidia M2090**
- **Keeneland CPU Core(s) – Intel Xeon X5660 (Westmere) @2.8GHz**
- **TitanDev CPU Core(s) – AMD Opteron 6200 (Interlagos) @2.6GHz**

Future Work & References:

Our primary goal is to now apply the Uintah Unified runtime system to the Alstom Boiler problem at full scale on the DoE Titan system using all CPU cores and GPUs

1. *Alan Humphrey, Qingyu Meng, Martin Berzins and Todd Harman. "Radiation Modeling Using the Uintah Heterogeneous CPU/GPU Runtime System", In Proceedings of the 2012 XSEDE Conference, 2012 ACM*
2. *Qingyu Meng, Martin Berzins, and John Schmidt. "Using Hybrid Parallelism to Improve Memory use in the Uintah Framework" In Proceedings of the 2011 TeraGrid Conference, 2011 ACM*
3. *Qingyu Meng and Martin Berzins. "Scalable Large-scale Fluid-structure Interaction Solvers in the Uintah Framework via Hybrid Task-based Parallelism Algorithms" Submitted to Concurrency and Computation: Practice and Experience, 2012*

www.sci.utah.edu