An Integrated Approach to Scaling Task-Based Runtime Systems for Next Generation Engineering Problems

Alan Humphrey
Brad Peterson
John Schmidt
Martin Berzins
{ahumphrey,bpeterson,jas,mb}@sci.utah.edu
SCI Institute
University of Utah
Salt Lake City, Utah 84112

Derek Harris
Ben Isaac
Jeremy Thornock
Todd Harman
{derekhar,bisaac,jthornoc,harman}@sci.utah.edu
Institute for Clean And Secure Energy
University of Utah
Salt Lake City, Utah 84112

Sidharth Kumar
Steve Petruzza
Allen Sanderson
Valerio Pascucci
{sidharth,spetruzza,allen,pascucci}@sci.utah.edu
SCI Institute
University of Utah
Salt Lake City, Utah 84112

ABSTRACT

The need to scale next-generation industrial engineering problems to the largest computational platforms presents unique challenges. Such problems may have complex coupled multi-physics models, as well as computationally and communication intensive algorithms not often seen in standard academic problems, while also needing high-demand I/O for analysis purposes. In such cases even codes with good existing scaling properties may need significant changes, addressed in a cross-cutting way to solve such problems at extreme scales. These challenges relate to system components that were not previously problematic on less complex problems and/or at smaller computational scales. This paper illustrates these challenges for Uintah, a highly scalable asynchronous many-task runtime system being applied to the modeling of a 1000 MWe ultra-supercritical coal boiler. In order to model this formidable, production engineering problem, not only was an integrated approach needed that built upon existing scalable components in areas such as complex stencil calculations and linear algebra, but the significant challenges of high demand I/O, complex task graphs, and infrastructure inefficiencies also had to be addressed. This integrated approach allowed this problem to run on 256K CPU cores on Mira, with good weak scaling to 512K CPU cores and similar strong scaling, except for radiation. The need for strong scaling of a new ray tracing-based radiation model required substantial Uintah infrastructure improvements before 119K CPU cores and 7.5K GPUs on Titan could be used, with scaling out to 256K CPU cores and 16K GPUs. The resulting code demonstrates not only excellent overall scalability but a significant improvement in overall performance for the full-scale, production boiler problem.

KEYWORDS

Uintah, AMT Runtime, Scalability, GPU, PIDX, Coal Boiler, Radiative Heat Transfer, Titan, Mira

ScalA'17, Denver, CO USA

DOI: 10.475/123_4

1 INTRODUCTION

The exponential growth in High Performance Computing (HPC) over the past 20 years has fueled a wave of scientific insights and discoveries, many of which would not be possible without the integration of HPC capabilities. This trend is continuing, with the DOE Exascale Computing Project [12] listing 25 major applications focus areas [14] in energy, science, and national security missions. The primary challenge in moving codes to new architectures at exascale is that while present codes may have good scaling characteristics on some present architectures, those codes may likely have components that are not suited to the extreme scale of new computer architectures, or to the complexity of real world applications at exascale. These challenges for example may involve potentially billion-way concurrency, multiple levels of heterogeneity (at both hardware and software levels) with multi-level memories, and a proposed target power ceiling of 20-40 megawatts (MW) for 1 exaflop, likely leading to power capping, non-uniform node-level performance and diminishing memory bandwidth and capacity relative to FLOP count. The same bandwidth limitations also apply to the I/O system at nearly all levels. The challenge of resilience is not well understood on architectures that are not yet defined. Nevertheless the possibility of more frequent faults leads to consideration of practical resilience strategies [15]. The complexity of these next generation problems imposes challenges in that the algorithms and computational approaches used will need to be considered to achieve scalability.

The intention in this paper is to demonstrate this process by considering a scaled-down exascale problem and show how the highly scalable Uintah asynchronous many-task (AMT) runtime system was adapted and used in a Large-Eddy Simulation (LES) to predict the performance of a commercial, 1000 megawatt electric (MWe) Ultra-Super Critical (USC) coal boiler. This problem has been considered as an ideal exascale candidate given that the spatial and temporal resolution requirements on physical grounds give rise to problems between 50 and 1000 times larger than those we can solve today.

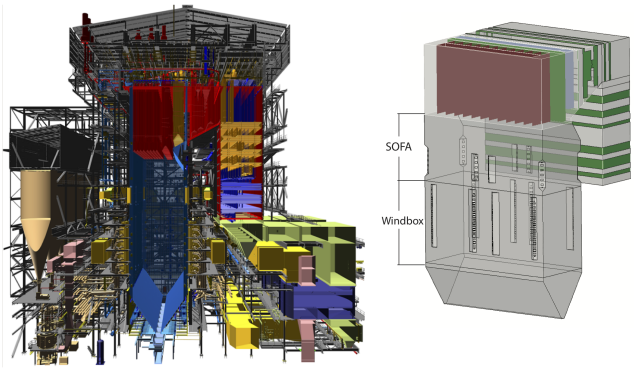
AMT codes like Uintah are attractive at petascale and exascale as the runtime approach shelters the application developer from the complexities introduced by future architectures. Uintah's approach using a directed acyclic graph (DAG) to represent the computation and associated dependencies has the advantage that the tasks in the task-graph may be executed in an adaptive manner, where choosing

an alternate task may avoid communications delays [11], effectively overlapping communication and computation. Uintah scales well to 768K CPU cores on standard stencil and particle multiphysics calculations [1]. The challenge addressed here is to see if this approach works on a complex real-life industrial problem, the target of a 351 million processor hour INCITE award.

In what follows, Section 2 describes the target boiler problem, gives a description of the Uintah framework and Arches turbulent combustion simulation component (used in this work), and provides discussion on simulation methodology and computational approaches used. Section 3 describes initial scalability and performance studies and shows that the stencil parts of the boiler simulation scale up to 512K CPU cores on Mira. A major challenge that is revealed is the scaling of the I/O approach that Uintah uses and this is addressed in Section 4. This allows production work to be undertaken on Mira as described in Section 5. The discrete ordinates radiation model approach used in the Mira production runs does not strong scale, and so a novel ray tracing approach to radiation modeling [3] that has shown promise on benchmark problems is applied to a production boiler problem to run on Titan with GPUs. Extensive design changes, described in Section 6, within the Uintah runtime were necessary, as the complexity of the boiler geometry and the all-to-all nature of radiation communication required a substantial rewrite in the way task-graphs are used. This approach to radiation allowed the calculation to run on 119K CPU cores and 7.5K GPUs simultaneously on Titan for this challenging problem, and to scale to 256K CPU cores and 16K GPUs. The conclusions, as given in Section 7, are that this new generation of problems needs scalability at every level, and this comes through integration of runtimes, linear solvers, I/O, computational algorithms, programming models, resiliency, and intelligent approaches to task graph representation and dependency analysis of these graphs, particularly when globally coupled problems such as radiation are considered. If one of these does not work at scale, the problem cannot be solved.

2 TARGET BOILER PROBLEM

GE Power is currently building new coal-fired power plants throughout the world. Many of these units may potentially be 1000 MWe, twin-fireball (no dividing wall) USC units, each providing power for nearly 1 million individuals. An example plant is shown in Figure 1a. Historically, twin-fireball (or 8-corner) units became part of the GE Power product offering because of the design uncertainty in scaling 4-corner units from a lower MWe rating to a much higher MWe rating. In order to decrease risk, both from GE Power's viewpoint, as well as from the customer's viewpoint, two smaller units were joined together to form a larger unit. The GE/Alstom power USC "flagship" boiler produces 1090 MWe. The geometrical complexity of the boiler is considerable with dimensions of 65m x35m x 15m and 430 inlets. The boiler has division panels, plates, super-heaters and re-heater tubing with about 210 miles of piping walls, and tubing made of 11 different metals with varying thickness In order to run the boiler needs a feed of 130 kg/s of coal (100 train cars of coal per day), and the O_2 from 1000 kg/s of air. The modeling challenges starting from the boiler blueprints were considerable for the combustion modelers. Both the 8-corner units and the 4-corner units have different mixing and wall absorption characteristics that must be fully understood in order to mitigate risk and have confidence in their respective designs. One key to this understanding is how the separated over-fired air (SOFA) inlets (Figure 1b) that inject pulverized coal and oxygen into the combustion chamber should be positioned and oriented and what effect these positions have on the heat flux distribution throughout the boiler.



(a) Entire unit. **(b) Primary wind-box and SOFA locations**

Figure 1: CAD rendering of GE Power's 1000 MWe USC two-cell pulverized coal boiler.

With a high-fidelity simulation tool, boiler designers can design, test, and optimize new boiler designs with relatively little investment. The simulation tool thus offers a platform for implementing and testing new technologies for more rapid deployment at a scale sufficient to augment or even displace existing energy options. The Carbon-Capture Multidisciplinary Simulation Center (CCMSC), has enabled the development of the advanced simulation science needed for these coal combustion systems. Using the Uintah Framework [17] as its primary simulation tool, the CCMSC has a two-fold mission (1) advancing simulation science beyond petascale (eventually to exascale) as well as VUQ-predictivity in real engineering systems, and (2) using high-performance computing and predictive science to achieve a societal impact by participating in the engineering design and startup of Advanced Ultra-Super-Critical (AUSC) oxy-coal combustion technology for an industrial partner, GE Power.

2.1 Uintah Framework

Uintah [17] is a software framework consisting of a set of parallel software components and libraries that facilitate the solution of partial differential equations on structured adaptive mesh refinement (AMR) grids. Uintah currently contains four main simulation components: 1.) the multi-material ICE code for both low and high-speed compressible flows; 2.) the multi-material, particle-based code MPM for structural mechanics; 3.) the combined fluid-structure interaction (FSI) algorithm MPM-ICE; and 4.) the Arches turbulent reacting CFD component that was designed for simulating turbulent reacting flows with participating media radiation. Separate from these components is an AMT runtime system.

Uintah decomposes the computational domain into a structured grid of rectangular cuboid cells. The basic unit of a Uintah simulation's Cartesian mesh (composed of cells) is termed a *patch*, and simulation variables that reside in Uintah's patches are termed *Grid* or *Particle* variables. The Uintah runtime system manages all of the complexity of inter-nodal data dependency (MPI), node-level parallelism, data movement between CPU and GPU and ultimately task scheduling and execution that make up a computational algorithm. The central idea is to use a graph representation of the computation to schedule work, as opposed to, say, a bulk synchronous approach in which blocks of communication follow blocks of computation. This graph-based approach allows tasks to execute in a manner that efficiently overlaps communication and computation.

Achieving performance with such an approach is not guaranteed, as the Uintah software has evolved to use dynamic execution of the task graph including out-of-order execution [1]. In order to address the need to reduce global memory usage with the larger core counts per node available on Titan and Mira along with the need to be able to run tasks on one or more accelerators per node (Titan), Uintah has moved to a nodal shared memory model [9] based on a combination of MPI+Pthreads and uses one MPI process per compute node. This design leverages MPI_THREAD_MULTIPLE, with each thread making its own MPI calls. This design is easily extendable to one MPI rank per NUMA region or per GPU, as will be seen on the upcoming Summit and Sierra systems. This thread-based approach has 1.) Decentralized execution [9] of the task-graph and is implemented by each CPU core requesting work itself, 2.) a lock-free shared memory approach [9] is implemented using atomics, allowing efficient access by all cores to the shared data on a node, and 3.) Accelerator task execution on a node is implemented through an extension of the runtime system that enables tasks to be executed efficiently (through preloading of data) on one or more accelerators per node. Using this hybrid approach also allows for improved load balancing, as only nodes need to be considered, not individual cores.

2.2 Arches - Combustion Component

The Arches component, which was designed for the simulation of turbulent reacting flows with participating media, is a three-dimensional, large eddy simulation (LES) code that uses a low-Mach-number, variable-density formulation to simulate heat, mass, and momentum transport in reacting flows. The LES algorithm solves the filtered, density-weighted, time-dependent coupled conservation equations for mass, momentum, energy, and particle moment equations in a Cartesian coordinate system [6]. This set of filtered equations is discretized in space and time and solved on a staggered, finite volume mesh. The staggering scheme consists of four offset grids, one for storing scalar quantities and three for each component of the velocity vector. Stability preserving, second order explicit time-stepping schemes and flux limiting schemes are used to ensure that scalar values remain bounded. Arches is second-order accurate in space and time and is highly scalable through Uintah and its coupled solvers like hypre to 256K cores [16].

2.3 Radiation Modeling

The USC electric power-generation boiler transfers thermal energy, generated by fuel oxidation, into steam which drives a turbine to generate electricity. The critical variable of interest for all boiler simulations is the heat flux to the surrounding walls. The dominant mode of heat transfer in the firebox of a coal-fired boiler is radiation, currently one of the most challenging problems in large-scale simulations, due to its global, *all-to-all* nature [3]. Firebox designs utilizing USC air-combustion technology require accurate radiative heat flux estimates in environments with increased CO_2 concentrations, higher temperatures, and different radiative properties for new metal alloys. Accurate radiative-heat transfer algorithms that handle complex physics are inherently computationally expensive [5]. Current simulation experience shows that radiative heat flux calculations take 1/3 to 2/3 of the total computational cost in boiler simulations. The heat transfer problems arising from the clean coal boilers involves solving the conservation of energy equation and radiative heat transfer equation (RTE) simultaneously. Thermal radiation in the target boiler simulations is loosely coupled to the computational fluid dynamics (CFD) due to time-scale separation.

The RTE, shown by Equation 2 in [3], represents the net radiative source term in the conservation of energy equation. The energy equation is conventionally solved by Arches (finite volume) and the temperature field, T in the energy equation [3], is used to compute net radiative source term. This net radiative source term is then fed back into the energy equation (for the ongoing CFD calculation) which is solved to update the temperature field, T.

2.3.1 Discrete Ordinates Method. The Discrete Ordinates Method, a modeling method developed at LANL for neutron transport is used to solve the RTE. The integration of DOM in a large scale CFD application is described in [7]. The solution procedure solves the RTE over a discrete set of ordinates and, like the pressure equation, is formulated as a linear system that is solved using *hypre* [7]. While this method works well, the performance of the algorithm is a function of the number of discrete ordinates (more ordinates, higher accuracy). Strong scalability suffers due to the use of the hypre linear solver as will be shown seen in Section 3 using Mira, where DOM was employed using 80 ordinate directions with system sizes on the order of 40 billion unknowns.

2.3.2 Reverse Monte Carlo Ray Tracing (RMCRT). Our recently developed RMCRT technology described in [3], which is amenable to GPU parallelization and shown to scale to 16K GPUS [4] was used for the production calculation on Titan. RMCRT algorithms have the potential to be more accurate and efficient [18] for solving the radiative flux when spectral participating media are present and is one of the few numerical techniques that can accurately solve for the radiative-flux divergence while accounting for the effects of participating media, naturally incorporates scattering physics, and lends itself to scalable parallelism [3]. The process is considered "reverse" through the Helmholtz Reciprocity Principle, e.g. incoming and outgoing intensity can be considered as reversals of each other [3].

In this approach, radiative properties are replicated on each node and ray tracing takes place without the need to pass ray information across node boundaries as rays traverse the computational domain. The principal challenge with RMCRT is an all-to-all communication phase introduced by this replication. Thus for a single, fine mesh approach with N_{total} mesh cells, the amount of data communicated is $O({N_{total}}^2)$. While accurate and effective at lower core counts, the volume of communication is untenable at large core counts. To

address this challenge, a strong-scalable, multi-level, mesh refinement approach was developed for both CPU [3] and GPU [4], in which a fine mesh is only used close to each grid point and a successively coarser mesh is used further away, significantly reducing MPI message volume and overall computational cost. The hybrid memory approach [9] of the Uintah computational framework also helps as only one copy of radiative properties/temperature is needed per multicore node or GPU. This algorithm allows for the radiation computation to be performed on a mesh resolution commensurate with the level of accuracy needed for this component of the physics, yet couples with other physics components, like the LES CFD, particle transport and particle reactions. This balanced approach to coupling multi-physics is made possible by Uintah's adaptive mesh and asynchronous execution policies that allow RMCRT to be used at this scale for the first time.

3 INITIAL SCALING EXPERIMENTS - MIRA

The Uintah Computational Framework has had a long history [1, 9] of large-scale, long running simulations. This history, with software that scaled well at 100K cores on similar cases would suggest that performing production level simulations that were 5X larger would merely require an increase in core count. However, the demands of running full-scale CCMSC boiler problems on a large fraction of current machine resources such as Titan and Mira have presented significant scaling challenges and complexities. These challenges have necessitated innovations throughout the entire computational pipeline, including algorithms, I/O, visualization and Uintah infrastructure improvements, as well as an approach to performance portability and a practical resilience strategy.

3.1 Initial Scalability Study

While the scalability of the Arches algorithm and Uintah was demonstrated in general for simpler systems, the large-scale boiler problem uncovered issues that, until resolved, prevented the completion of the production runs. A simpler representation of the production level geometry was used to perform a weak scaling study for multiple timesteps that included both computations and I/O on the DOE Mira system. The results of this study can be seen in Figure 2, where 1.) the black line shows the standard timestep with the Pressure-Poisson solve, 2.) the red line shows the timestep with the radiation solve using Discrete Ordinates, and 3.) the green line shows poor scalability of the output using the original native Uintah I/O subsystem.

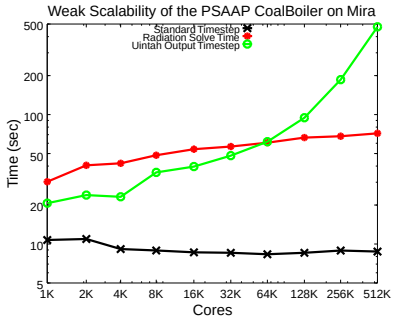


Figure 2: Weak scalability of the coal boiler simulation running on Mira from 1K to 512K cores.

The weak scaling study indicated significant challenges with the native Uintah I/O subsystem, and that any timestep performing I/O including both data dumps and checkpoint/restart were severely constrained by native Uintah I/O. This I/O system had been in place since the time when 2K core systems were considered large, and directories with several thousand files was not unusual. Uintah's I/O system would generate two files per MPI rank. However, as system sizes increased by several orders of magnitude, the I/O design of Uintah never was critically examined for previous production simulations (<100K cores). Figure 2 shows that the first major scalability challenge was to address the scalability of I/O.

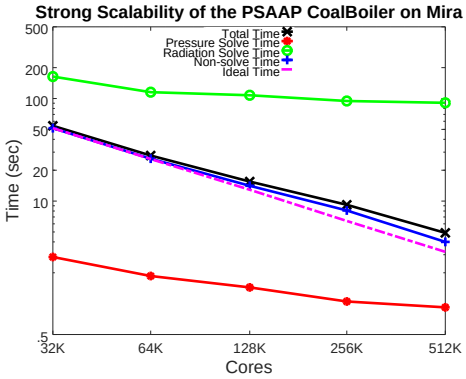


Figure 3: Strong scalability of the coal boiler simulation running on Mira from 32K cores to 512K cores.

A strong scaling study was then performed to demonstrate that the overall Arches algorithm strong scaled to near full machine capacity of Mira. In Figure 3, 1.) the black line shows the scalability of an entire time step (overall Arches algorithm) that includes the pressure-poisson solve using hypre [2], 2.) the blue line shows the scalability of the time step without the pressure-poisson solve, 3.) the red line shows the scalability of the pressure-poisson solve, and 4.) the green line illustrates the poor strong scalability of the Discrete Ordinates method for radiation using hypre [2].

The Arches algorithm [6, 16] has a Pressure-Poisson linear solve (10^8 unknowns) along with a Discrete Ordinates linear solve (10^{10} unknowns) as well as other computational components that do not require linear solves. The strong scalability of the overall Arches algorithm nearly follows the idealized scalability shown in the dashed purple line in Figure 3. The linear solve for the Pressure-Poisson element in the Arches algorithm does not severely impact the overall strong scalability for the production coal-boiler case [16]. These results suggest that 1.) it is important to profile a full, representative case that adequately represents the complex nature of the production case geometry, and 2.) include I/O in any scalability studies, as it can be a limiting factor in the amount and frequency of data that may be generated for any given simulation.

4 PIDX INPUT AND OUTPUT SCALABILITY

With the Uintah Data Archive (UDA) I/O subsystem, every node writes data for all of its cores into a separate file. Therefore on Mira, there is one file for every 16 cores (16 cores per node). This form

of I/O is an extension to the file-per process style of I/O commonly adopted by many simulations. There is also an XML based meta-data file associated with every data file that stores type, extents, bounds, and other relevant information for each of the different fields. For relatively small numbers of core counts, this I/O approach works well. However, I/O performance degrades significantly for simulations with several hundreds of thousands of patches/processors. The cost of both reads and writes for large number of small files becomes untenable.

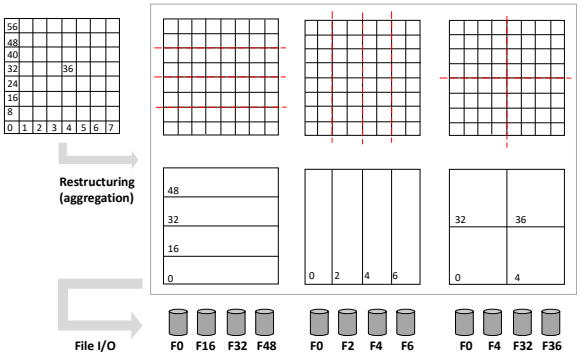


Figure 4: Schematic diagram of deployed PIDX I/O scheme. Three variations of restructuring shown with restructure along the Z-slab following the rank mapping of the application.

In the current production boiler cases, a modified version of the PIDX I/O library [8] was used. PIDX was tuned to use a customized two-phase I/O approach to write data in parallel. The first phase involves restructuring of simulation data into large blocks while preserving the original multidimensional row-order format. This phase is followed by actual disk-level I/O writes (second phase) by the processes holding the restructured data. By adopting this two-phase I/O, the shortcomings of small data reads were mitigated while providing the benefit of sub-filing shown in Figure 4.

4.1 Weak scaling results

The weak scaling performance of the I/O system was evaluated when writing data for a representative Uintah simulation on Mira. In each run, Uintah wrote out 5 timesteps consisting of 72 fields (Grid Variables). The simulation dumps data at every 10th time-step. The patch size for simulation was 12x12x12. The number of cores were varied from 7920 processes to 262,890. Looking at the performance results in Figure 5, the PIDX I/O system scales well for all core counts and performs better than original Uintah UDA formatted I/O. The PIDX I/O system demonstrates almost linear scaling up to 262,890 cores whereas performance of UDA I/O starts to decline after 16,200 cores. At 262,890 cores, the PIDX I/O system achieves an approximate speedup of 5.3X over UDA I/O.

PIDX was then used in both Mira production boiler cases. Both simulations were carried out at 260K cores. Due to the performance benefits offered by PIDX, scientists were able to write data at much higher frequency. In order to complete the end-to-end pipeline, VisIt was installed on the Argonne visualization system Cooley, with the PIDX plugin reader to visualize the data generated from these large

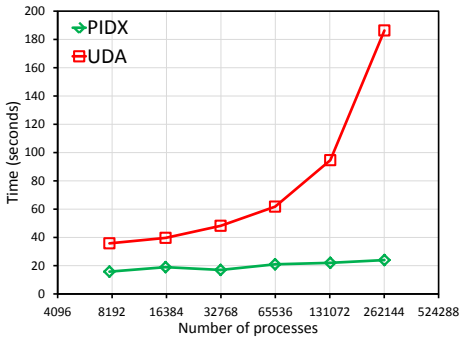


Figure 5: Production run results.

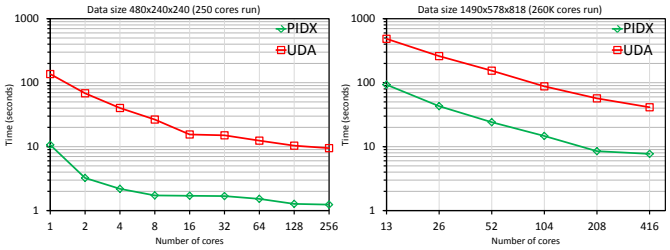


Figure 6: VisIt reading time for a dataset generated by runs with 250 cores (L) run and a 260K cores (R).

scale simulations. In terms of outputs, close to 200 terabytes of data was written, requiring only 2% of the entire simulation time, compared to nearly 50% of the time if the data were written in the native UDA format.

4.2 Data Input and Visualization

The PIDX library was also used for reading the simulation data for verification and visualization purposes. For the visualization a VisIt reader plug-in was developed that dramatically improved performance and user experience.

In the former Uintah UDA format, data was represented by a collection of mesh patches represented by individual files. The number of MPI ranks determined the number of individual files that were written to disk. Since the size of the mesh patches is relatively small (12^3), the total number of patches will increase as the dataset size grows. This creates a reading size bottleneck that was solved by computing a domain decomposition that creates a number of patches equal to the number of cores available. This allows for efficiently reading the same datasets from different core counts, thus enabling scalability. Figure 6 shows the performance scaling results for the new PIDX reader against the former UDA format reader. These performance results demonstrate that using the PIDX reader the user needs only a few cores to access very large datasets in contrast to the very large core counts required with the older Uintah reader.

5 MIRA PRODUCTION CASES - RESULTS

In making use of the improvements to scalability of the entire code, two production cases were considered using the geometry, inlet

parameters and operating parameters of a GE Power 8 Corner Unit. The first case represented the operation of the commercial unit that is currently in production, whereas the second case represented alterations to the inlet parameters to investigate a more uniform energy distribution. Each case was run for approximately 20 seconds of physical time which is considered sufficient for the boiler to achieve a steady state distribution.

Table 1: Computational Aspects of 8-corner Boiler Simulations

Item	Case 1	Case 2
Time/timestep	4.5 s	3.0 s
Pressure Solve	1.9 s	1.0 s
Radiation Solve	101 s	79.3 s
Data to Disk	5.5 min	33.2 s
Data Dumps	77	1030
Data Size	9.9 TB	180 TB
# TimeSteps	236,500	220,979
Simulated Time	19.38 s	17.92 s
CPU Hours	97 M	110 M

Table 1 shows the computational aspects of the 2 cases that were run on Mira, simulating the 8-corner unit. Each production case was run at 260K cores with 455M grid cells at a resolution of 4.25cm with 16 MPI ranks per node using a timestep of $8e^{-5}$ and about 400MB of memory per rank. Table 1 shows that between the first and second cases additional speed-ups were achieved in the pressure solve due to work being done in Uintah/Arches. The most significant difference was the switch to PIDX for I/O yielding 33 second write times compared to the 5.5 minutes write times for Case 1 using legacy Uintah I/O code at this scale. Ultimately, Case 2 wrote 1030 datasets allowing for the creation of 3D rendered movies of the case. In each case on Mira the default Uintah task scheduler was used, which uses 1 MPI rank per core (16 MPI ranks per node).

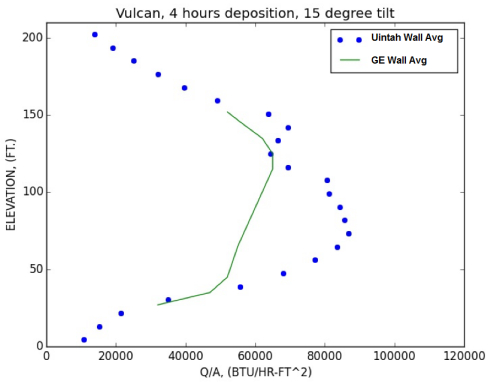


Figure 7: Heat absorption profile as a function of the elevation.

Though validation of the simulation data against experimental data was performed, the proprietary nature of both the simulation and experimental data makes publication of these comparisons problematic. However, working closely with the GE Power engineers made it possible to validate the results of these simulations against their previous results. Figure 7 shows the heat absorption profile (x-axis) as a function of the elevation in the boiler (y-axis). The solid green line shows GE Power's wall-averaged absorption profile tentative estimates for the expected operating conditions in the unit. The blue dots show the average absorption profile computed from Case 1. The average absorption profile predicted in Case 1 is different from the tentative estimates due to the higher fidelity modeling performed with Arches, but it is in relatively good agreement with the actual absorption profile based on discussions with GE Power engineers and the existing proprietary data provided. The second case was run with changes to the inlet geometry parameters to optimize gas-side energy imbalance (GSEI) by changing the flow pattern in the wind-box as well as the SOFA inlets.

The key result from this work is the confidence that has been established with GE Power to demonstrate that high resolution LES simulations are a useful tool for exploring a range of operating conditions with the potential to be used for future designs. This is the first time that computational design at this scale has been used for such a complex combustion problem with petascale simulations. Future studies for the unit will investigate design and operation adjustments to achieve incremental improvements in gas-side energy imbalance. GE will consider testing the new conditions in the existing unit when significant improvements are discovered.

6 TITAN PRODUCTION CASE

The principal goals of the Titan case were advances in simulation methodology, proving the viability of our experimental RMCRT approach to radiation, and advancing the development within the Uintah infrastructure to efficiently handle globally coupled problems involving radiation in heterogenous environments. As part of the infrastructure development, Uintah's multi-threaded, heterogeneous task scheduler [10] was employed on Titan, which uses a shared memory model on node (a combination of MPI+Pthreads+CUDA), with 1 MPI rank per node and individual threads executing tasks and managing data movement to-and-from Titan's nodal GPUs. The Mira 8-corner boiler case (Case 1) was used on Titan, replacing DOM with a GPU-based RMCRT algorithm [4] for radiation. The Titan production case ran using 7467 nodes (119,472 CPU cores), as opposed to the equivalent computation with 260,712 CPU cores on Mira. This reduction in node count (approximately half of Titan) was to mitigate the machine instabilities and sustained node failures that became key motivations for pursuing the resilience strategy being developed for Uintah by Sahasrabudhe, et al. [15].

The production calculation used a patch size of 12^3, optimal for the ongoing CPU-based CFD computation. However, this meant that the GPUs were initially underutilized in comparison to earlier benchmark problems [4] that used significantly larger patch sizes to increase GPU workload. This issue is addressed in [13] by leveraging NVIDIA CUDA Streams, enabling concurrent execution of multiple GPU kernels and maintaining the optimal patch size for the overall simulation. This design improvement has yielded an approximately 1.4X speedup when using the GPUs vs the CPUs for the RMCRT radiation calculation on 7467 GPUs.

6.0.1 Infrastructure Improvements. Two major infrastructure changes were made to Uintah. First, the adoption of the C++11 standard allowed the removal of tens of thousands of lines of code

previously used to provide functionality now available through the standard library, through portable synchronization primitives, atomics and other concurrency offerings. Use of C++11 move semantics have also led to the development of novel lock-free data structures, providing Uintah with a portable approach to multi-threading absent the burden of maintaining complex and error-prone code.

While Uintah with the RMCRT algorithm scaled well with a benchmark problem on both CPUs [3] and GPUs [4], the geometrical and task complexity of the full boiler simulations on Titan required further changes to make Uintah's task dependency analysis phase (part of task graph creation) scale when global data dependencies arise in large problems.

6.0.2 Multiple Task Graphs. Within Uintah, analyzing tasks for intra- and inter-nodal data dependencies is referred to as **dependency analysis**, part of the task graph compilation process. For standard stencil calculations, where each compute node only needs to search surrounding nodes for neighboring patches, dependency analysis completes in milliseconds, even at scale. However, with the introduction of global dependencies, initial boiler runs on Titan required 4.5 hours for this dependency analysis at production scale. Additionally, the simulation required alternating between task execution patterns for time steps involving either 1.) the standard computational fluid dynamics (CFD) calculation or 2.) CFD plus a radiation calculation to recompute the radiative source term (on Titan's GPUs) for the ongoing CFD calculation. Alternating between these separate task execution patterns occurred every 20 time steps and required reanalysis of all global dependencies for the radiation solve, incurring potentially another 4.5 hour dependency analysis. As the dependency analysis is required for automated MPI message generation, our previous strategy on small problems was to use a simpler task graph for the CFD timesteps and then to recompile the task-graph on radiation timesteps. The cost of recompilation for the full boiler problem made this unworkable.

The solution was to add support within Uintah for multiple primary task graphs: one for the ongoing CFD calculation and one for CFD timesteps with a radiation calculation to update the radiative source term. With multiple task graphs, this repeated dependency analysis phase was eliminated by generating both the CFD and CFD+radiation task graphs once during initialization and cached for subsequent reuse throughout the remainder of the simulation. Uintah's Simulation Controller was modified to manage passing the appropriate task graph to the task scheduler at each timestep [13].

6.0.3 Multiple Processor Neighborhoods. For each compute node, Uintah generates a local task graph for tasks residing on patches owned by that node, and the resulting data dependencies for automated MPI message generation. The Uintah load balancer then creates a "processor neighborhood" for halo exchange. Previously, the task with the maximum number of halo layers designated the halo length for the neighborhood. For the target production problem, this naive approach was no longer viable for three reasons, 1.) a large halo number due to the global nature of radiation calculations, 2.) non-uniform halo requirements across AMR mesh levels, and 3.) applying this large halo number to almost 1000 tasks per timestep resulted in massive, exaggerated communication dependencies. Figure 8 illustrates these differing halo requirements. On a single patch, two distinct tasks may have different halo requirements, thus requiring separate data dependency analysis. Our solution has been to split tasks into two processor neighborhoods, one for local halo exchange and another for the non-local halo requirement. This approach reduced the task-graph compile times by 13.5x at 128K cores for a 2-level mesh problem, bringing the initial 4.5 hour dependency analysis phase down to 20 minutes. Work is currently underway to significantly reduce this remaining time through parallel task graph construction and elimination of additional complexities involved with task graph representation (see [13]).

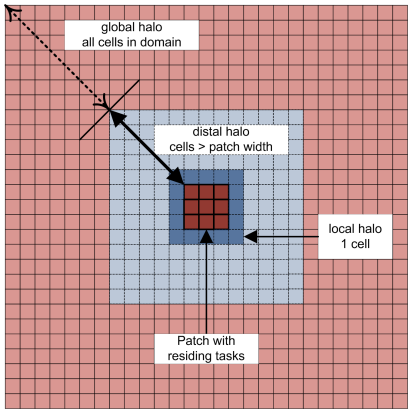


Figure 8: Local, distal, and global halo requirements

6.0.4 RMCRT Results. After addressing the scaling challenges arising from the most complicated geometry that the Uintah Framework has ever attempted to simulate, and the accompanying infrastructure changes, excellent strong scalability is achieved for the full production boiler case when using RMCRT for the radiation calculation. Table 2 shows the mean time per timestep (averaged over 10 timesteps involving a radiation solve) from 16K to 256K CPU cores (1k to 16K GPUs respectively). These results show strong scaling out to near the full extent of Titan for a full production boiler case with Arches and RMCRT, and are the first such results of such a complex geometry coupled to a full combustion model. Additionally, a comparison of the RMCRT results against the standard discrete ordinates method was made. Figure 9 shows a comparison of the instantaneous divergence of the heat flux, one of the primary quantities of interest in our boiler simulations, computed using the CPU (left) and GPU (right) implementations of multi-level RMCRT. In each view three slices through the interior of the computational domain are shown. These simulations were run on 16K Titan cores using 1024 GPUs and the qualitative agreement is excellent. This is first production simulation of a coal fired boiler utilizing RMCRT to solve the RTE and shows that RMCRT is a viable radiation approach for accelerator-based architectures such as Titan and the upcoming Summit system, unlike the discrete ordinates method, used on Mira.

Table 2: Strong Scaling Results: Full Boiler with GPU-RMCRT

Cores/GPUs	16k/1k	32k/2k	64k/4k	128k/8k	256k/16k
Time (sec)	821.13	407.31	202.69	99.39	55.06

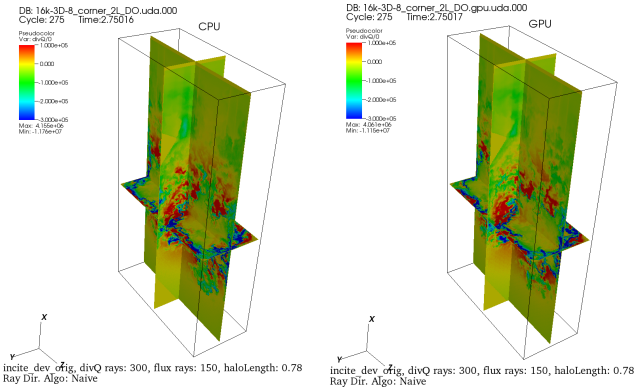


Figure 9: Comparison of the instantaneous divergence of the heat flux for multi-level RMCRT, CPU (left) and GPU (right).

7 CONCLUSIONS

This work has introduced an excellent exascale candidate problem through the simulation of a commercial, 1000 MWe Ultra-Super Critical (USC) boiler, the largest currently in production worldwide, using Large-Eddy Simulation (LES) predictions, requiring 351 Million CPU hours on Mira and Titan. The overall objective of this work was in understanding how we can solve such a problem through the use of an AMT runtime with scalability improvements in the runtime itself, linear solvers, I/O, and computational algorithms. We have demonstrated that using a DAG approach within an AMT runtime is critical for computational tasks to be executed in an adaptive manner, effectively overlapping communication and computation.

Through this work, we have exposed areas even within an advanced, scalable AMT runtime system that need careful design consideration for post-petascale and eventually exascale platforms, particularly when globally coupled problems such as radiation are considered. A key lesson this work conveys is that the success of large, production-scale simulations depends upon scalability at every level. If any single component within the simulation pipeline does not scale, the problem cannot be solved. It is through the integration of these scalable components and subsystems that the next generation of problems may be solved on exascale systems.

Our results have demonstrated the potential role that LES simulations can have on analysis and design of an operational commercial boiler and that simulations can be used as a design tool for future systems, and that choosing hardware appropriate algorithms such as RMCRT is important in achieving scalable results. Finally, to achieve the results shown in this work for production level petascale computations (256K Mira CPU cores, 119K Titan CPU cores/7.5K GPUs), significant code and algorithmic innovations were required including the use of the novel PIDX I/O library that achieved a nearly order of magnitude improvement in I/O performance, and fundamental changes to Uintah's dependency analysis phase.

8 ACKNOWLEDGEMENTS

This material is based upon work supported by the Department of Energy, National Nuclear Security Administration, under Award Number(s) DE-NA0002375. An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under contract DE-AC02-06CH11357. This research also used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. We would like to thank all those involved with Uintah past and present.

REFERENCES

[1] M. Berzins, J. Beckvermit, T. Harman, A. Bezdjian, A. Humphrey, Q. Meng, J. Schmidt, , and C. Wight. 2016. Extending the Uintah Framework through the Petascale Modeling of Detonation in Arrays of High Explosive Devices. *SIAM Journal on Scientific Computing* 38, 5 (2016), 101–122. DOI:https://doi.org/10.1137/15M1023270

[2] R. Falgout, J. Jones, and U.M. Yang. 2006. The Design and Implementation of hypre, a Library of Parallel High Performance Preconditioners. In *Numerical Solution of Partial Differential Equations on Parallel Computers*, AreMagnus Bruaset and Aslak Tveito (Eds.). Lecture Notes in Computational Science and Engineering, Vol. 51. Springer Berlin Heidelberg, 267–294.

[3] A. Humphrey, T. Harman, M. Berzins, and P. Smith. 2015. A Scalable Algorithm for Radiative Heat Transfer Using Reverse Monte Carlo Ray Tracing. In *High Performance Computing*, Julian M. Kunkel and Thomas Ludwig (Eds.). Lecture Notes in Computer Science, Vol. 9137. Springer, 212–230.

[4] A. Humphrey, D. Sunderland, T. Harman, and M. Berzins. 2016. Radiative Heat Transfer Calculation on 16384 GPUs Using a Reverse Monte Carlo Ray Tracing Approach with Adaptive Mesh Refinement. In *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 1222–1231.

[5] I. Hunsaker. 2013. *Parallel-distributed, Reverse Monte-Carlo Radiation in Coupled, Large Eddy Combustion Simulations*. Ph.D. Dissertation. Dept. of Chemical Engineering, University of Utah.

[6] J.Spinti, J. Thornock, E. Eddings, P.J. Smith, and A. Sarofim. 2008. Heat Transfer to Objects in Pool Fires. In *Transport Phenomena in Fires*. WIT Press, Southampton, U.K.

[7] G. Krishnamoorthy, R. Rawat, and P.J. Smith. 2004. Parallel Computations of Radiative Heat Transfer Using the Discrete Ordinates Method. *Numerical Heat Transfer, Part B: Fundamentals* 47, 1 (2004), 19–38.

[8] S. Kumar, J. Edwards, P.-T. Bremer, A. Knoll, C. Christensen, V. Vishwanath, P. Carns, J. A Schmidt, and V. Pascucci. 2014. Efficient I/O and storage of adaptive-resolution data. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press, 413–423.

[9] Q. Meng and M. Berzins. 2014. Scalable large-scale fluid-structure interaction solvers in the Uintah framework via hybrid task-based parallelism algorithms. *Concurrency and Computation: Practice and Experience* 26, 7 (May 2014), 1388–1407. DOI:https://doi.org/10.1002/cpe

[10] Q. Meng, A. Humphrey, and M. Berzins. 2012. The Uintah Framework: A Unified Heterogeneous Task Scheduling and Runtime System. In *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*. 2441–2448.

[11] Q. Meng, A. Humphrey, J. Schmidt, and M. Berzins. 2013. Investigating Applications Portability with the Uintah DAG-based Runtime System on PetaScale Supercomputers. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '13)*. ACM, New York, NY, USA, Article 96, 12 pages.

[12] U.S. Department of Energy. 2017. Exascale Computing Project. (2017). https://exascaleproject.org/.

[13] B. Peterson, A. Humphrey, J. Schmidt, and M. Berzins. 2017. Addressing Global Data Dependencies in Heterogeneous Asynchronous Runtime Systems on GPUs. In *Submitted - Third International Workshop on Extreme Scale Programming Models and Middleware (ESPM2)*. IEEE Press.

[14] J. Russell. 2017. Doug Kothe on the Race to Build Exascale Applications. (2017). https://www.hpcwire.com/2017/05/29/doug-kothe-race-build-exascale-applications/.

[15] D. Sahasrabudh, M. Berzins, and John Schmidt. 2017. Resiliency for Uintah Without Checkpointing. In *Preparing for Submission*.

[16] J. Schmidt, M. Berzins, J. Thornock, T. Saad, and J. Sutherland. 2013. Large Scale Parallel Solution of Incompressible Flow Problems using Uintah and hypre. In *Proceedings of CCGrid 2013*. IEEE/ACM.

[17] Scientific Computing and Imaging Institute. 2015. Uintah Web Page. (2015). http://www.uintah.utah.edu/.

[18] X. Sun and P. J. Smith. 2010. A parametric case study in radiative heat transfer using the reverse monte-carlo ray-tracing with full-spectrum k-distribution method. *Journal of Heat Transfer* 132, 2 (2010), 024501.