Fall Technical Meeting
of the Eastern States Section of the Combustion Institute
Hosted by the University of Connecticut, Storrs, CT
October 9-12, 2011

Computationally-Efficient Parallel Implementation of Combustion Chemistry in LES/PDF Computations

Varun Hiremath [1], *Steven R. Lantz* [2], *Haifeng Wang* [1], *Stephen B. Pope* [1]

[1] Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA
[2] Center for Advanced Computing, Cornell University, Ithaca, NY 14853, USA

Abstract

Large scale combined *Large-Eddy Simulation* (LES)/*Probability Density Function* (PDF) parallel computations of reactive flows with detailed chemistry involving large numbers of species and reactions are computationally expensive. Among the various techniques used to reduce the computational cost of representing chemistry, the three approaches in widest use are: (1) *mechanism reduction*, (2) *dimension reduction*, and (3) *tabulation*. In addition to these approaches, in large scale parallel LES/PDF computations, we need strategies to distribute the chemistry workload among the participating CPUs to reduce the overall wall clock time of the computations. Here we present computationally-efficient strategies for implementing chemistry in parallel LES/PDF computations using *in situ* adaptive tabulation (ISAT) and *x2f_mpi* – a Fortran library for parallel vector-valued function evaluation (used with ISAT in this context). To test the strategies, we perform LES/PDF computations of the Sandia Flame D with chemistry represented using the 16-species ARM1 reduced mechanism. We present three parallel strategies for redistributing the chemistry workload, namely (a) PLP, purely local processing; (b) URAN, the uniform random distribution of chemistry computations among all CPUs following an early stage of PLP; and (c) P-URAN, a Partitioned URAN strategy that redistributes the workload only among partitions or subsets of the CPUs. We show that among these three strategies, the P-URAN strategy (i) yields the lowest wall clock time, which is within a factor of 1.5 of an estimate for the lowest achievable wall clock time; and (ii) shows acceptable weak-scaling up to 9,216 cores.

1 Introduction

Computations of turbulent combustion flows using detailed chemistry involving a large number of species and reactions are computationally expensive. Modern chemical mechanisms of real fuels involve hundreds or thousands of species and thousands of reactions [1, 2]. Incorporating such detailed chemistry in the combustion flow calculations is computationally prohibitive. Among the various efforts put into reducing the computational cost of representing chemistry, the three approaches in widest use are: (1) *mechanism reduction* to reduce the number of species and reactions involved [3, 4, 5]; (2) *dimension reduction* to represent chemistry using a reduced number of variables [6, 7, 8, 9]; and (3) *tabulation* to significantly reduce the cost of expensive evaluations of the reaction mappings involving ODE integrations [10, 11, 12]. In recent times, combined methodologies have also been developed, wherein reduced reaction mechanisms or dimension reduction methods are used in conjunction with tabulation [13, 14].

Due to the high computational cost of turbulent combustion problems, most of the modern day simulations are performed in parallel on multiple CPUs using distributed computing. Thus, in addition to the aforementioned techniques, when performing large scale parallel LES/PDF computations, strategies are needed to efficiently distribute the chemistry workload among the participating CPUs to reduce the overall wall clock time of the computations [15, 16, 17, 18]. In this paper we present parallel strategies for the implementation of chemistry using *in situ* adaptive tabulation (ISAT) [10] and *x2f_mpi* – a Fortran library for parallel vector-valued function evaluation (used with ISAT in this context) [17]. The parallel strategies are tested by performing LES/PDF simulations of the Sandia Flame D with chemistry represented using the 16-species ARM1 mechanism.

The outline of the remainder of the paper is as follows: in Section 2 we describe our hybrid LES/PDF solver and the parallel strategies implemented using ISAT and *x2f_mpi* for implementing chemistry; in Section 3 we describe the computational details for simulating Sandia Flame D; in Section 4 we present performance results for different parallel strategies; and finally in Section 5 we state our conclusions.

2 Hybrid LES/PDF Solver

2.1 LES Solver

In this study we use an LES solver based on the Stanford LES code [19, 20]. The LES solver solves the Eulerian transport equations for mass, momentum and energy using finite difference methods on structured non-uniform grids. It supports both Cartesian and cylindrical coordinate systems; is second order accurate in space and time; and is parallelized (using MPI) by domain decomposition in two dimensions.

2.2 PDF Solver

We use the PDF solver, *HPDF*, developed at the Turbulence and Combustion Research Group at Cornell [21]. The HPDF solver has second-order accuracy in space and time; supports Cartesian and cylindrical coordinate systems; is parallelized (using MPI) by domain decomposition in two dimensions; and has a general interface to facilitate coupling with existing LES (or RANS) solvers. In HPDF, the thermo-chemical composition of the fluid within the solution domain is represented by a large number of particles.

The HPDF solver has three main components which account for:

1. *transport*: the motion of particles in the physical domain, including a random component to represent the effects of subgrid-scale turbulence;
2. *mixing*: the change in composition of a particle due to mixing with neighboring particles (which models the effects of molecular mixing); and
3. *reaction*: the change in composition of a particle due to chemical reaction.

These components are implemented in *fractional steps* using splitting schemes [22].

In this study we use the first-order $\mathbb{TMR}$ splitting scheme (which is found to perform as well as the second-order splitting scheme for jet flames [21]), which denotes taking fractional steps of *transport*, $\mathbb{T}$; *mixing*, $\mathbb{M}$; and *reaction*, $\mathbb{R}$ in this order on each time-step. The $\mathbb{KP}$ [23] SDE scheme is used to integrate the *transport* equations; and the *mixing* is represented using the modified Curl [24] mixing model. In the remainder of this section, we focus on the implementation details of the *reaction* fractional step.

2.3 Particle Representation

We consider a reacting gas-phase mixture consisting of n_s chemical species, composed of n_e elements. The thermo-chemical state of the mixture (at a given position and time) is completely characterized by the pressure p, the mixture enthalpy h, and the n_s-vector $\mathbf{z}$ of specific moles of the species. We consider an isobaric system with a fixed specified pressure p, and so the thermo-chemical state is fully characterized by $\{\mathbf{z}, h\}$. In the HPDF solver, the particles carry the composition $\{\mathbf{z}, h\}$.

2.4 Reaction Mapping

In the reaction fractional step, a particle's chemical composition $\mathbf{z}$ evolves (at constant enthalpy h) in time according to the following set of ordinary differential equations (ODEs)

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{S}(\mathbf{z}(t)), \tag{1}$$

where $\mathbf{S}$ is the n_s-vector of chemical production rates determined by the chemical mechanism used to represent the chemistry.

The *reaction mapping*, $\mathbf{R}(\mathbf{z}, t)$ is defined to be the solution to Eqn.1 after time t starting from the initial composition $\mathbf{z}$. The reaction mapping obtained by directly integrating the set of ODEs given by Eqn.1 is referred to as a *direct evaluation* (DE). We use DDASAC [25] for performing ODE integration.

2.5 Chemistry Solver

Owing to the large cost of direct evaluation of reaction mapping involving large number of species, *in situ* adaptive tabulation (ISAT) is used in the HPDF solver to reduce the cost of chemistry calculations. In addition, we use the *x2f_mpi* library to distribute the chemistry workload efficiently among the participating CPUs in large scale parallel LES/PDF simulations.

2.6 Domain Decomposition

The LES computations are performed on structured non-uniform grids in Cartesian or cylindrical coordinate system. We denote the grid used for LES computations by $N_x \times N_y \times N_z$ (in the three principal directions). In performing parallel LES/PDF computations (using the hybrid LES/HPDF solver) on N_c CPUs (or cores), the computational domain is decomposed into N_c sub-domains and each CPU performs the computations of one sub-domain. The domain decomposition is done in the first two principal directions, and is denoted by $D_x \times D_y$, where $D_x D_y = N_c$. The domain decomposition is done such that N_x and N_y are exact multiples of D_x and D_y, respectively. In addition, the domain decomposition in the LES solver is restricted by the grid size such that $D_x \leq N_x/2$ and $D_y \leq N_y/2$. The HPDF solver has the capability to use its own domain decomposition independent of the LES solver, but in the current study, we use the same domain decomposition in both the LES and HPDF solvers.

2.7 Parallel Strategies

In performing parallel LES/PDF computations with chemistry tabulation using ISAT, each CPU has its own ISAT table for tabulating the chemistry. At the reaction fractional step, the reaction mappings for all the particles in the computational domain need to be evaluated. A particle whose reaction mapping has been evaluated is called a *resolved* particle; and the act of resolving a particle by successfully retrieving a linear approximation to the reaction mapping from an ISAT table is called a *retrieve*. In parallel computations, a particle on a given CPU can be resolved in one of the following ways:

1. retrieve attempt from the local ISAT table;
2. retrieve attempt(s) from remote ISAT table(s);
3. direct evaluation (followed by addition to the ISAT table) on the local CPU;
4. direct evaluation (followed by addition to the ISAT table) on a remote CPU.

The goal is of course to resolve all the particles in the minimum possible wall clock time by redistributing the chemistry workload among all the CPUs, but this is not a trivial task because:

1. the time required to resolve a given particle is unknown ahead of time;
2. the time to resolve may vary by 4 orders of magnitude, as the retrieve time from an ISAT table is $\mathcal{O}(10)\mu s$ while a direct evaluation may take $\mathcal{O}(10^5)\mu s$;
3. furthermore, the probability of retrieve from an ISAT table depends on the history and duration of the run.

The *x2f_mpi* library is used as an interface between the HPDF solver (for the *reaction* fractional step) and the ISAT tables to redistribute the chemistry workload (of resolving all the particles) efficiently among the participating CPUs to reduce the overall wall clock time of the computations.

Here we present three parallel strategies implemented using the *x2f_mpi* library to redistribute the chemistry workload in the HPDF solver.

1. Purely Local Processing (PLP):

   In this strategy, all the particles on a CPU are resolved (i.e., the reaction mapping is evaluated) using the local ISAT table without any message passing or load redistribution. This in some sense is the same as invoking ISAT directly from HPDF on each CPU without using the *x2f_mpi* interface.

   The main pros of this strategy are:

   (a) ease of implementation;
   (b) no communication cost;
   (c) higher probability of retrieving particles from the local table;

   and some of the cons are:

   (a) load imbalance (between CPUs in the reactive zone and those in the coflow/air);
   (b) relatively high wall clock time.

2. Uniformly Random Distribution (URAN):

   This strategy aims at achieving ideal load balancing statistically by evenly distributing the chemistry workload among all the participating CPUs. The strategy involves one initial HPDF step of PLP to initialize the local ISAT tables. In the subsequent steps, on each CPU, a "quick try" is first made to attempt to resolve particles by retrieving from the local ISAT table; following this, there is a uniform random distribution of all the unresolved particles to all the CPUs. This strategy thus ensures that every CPU receives (approximately) the same number of particles to resolve, and with a similar distribution of particles from the reactive and non-reactive zones of the computational domain.

The main pro of this strategy is:

(a) close to ideal load balancing, after the initial "quick try" lookup;

and the cons are:

(a) relatively costly all-to-all communication;
(b) lower probability of retrieving particles (due to the random distribution of unresolved particles over all the cores);
(c) poor scaling (to large number of CPUs) due to all-to-all communication.

3. Partitioned Uniformly Random Distribution (P-URAN):

This is a new strategy which is a combination of the previous two, PLP and URAN, strategies. This strategy works in two stages: in stage 1 (for a specified duration of time) the PLP strategy is used to resolve particles; then in stage 2, the participating CPUs are partitioned into smaller groups, and within each partition the URAN strategy is used to uniformly distribute the chemistry workload among the CPUs in that partition.

The pros of this strategy are:

(a) relatively higher probability of retrieving from the local tables due to the initial PLP stage;
(b) reduced communication cost compared to URAN (communication restricted to within smaller partitions);
(c) good load balancing within partitions;
(d) good scaling to large number of CPUs.

and some of the cons are:

(a) load imbalance among different partitions;
(b) the need to determine additional parameters: (i) duration of the PLP stage; and (ii) size of the partitions.

To specify the parameters used in the P-URAN strategy, in the remainder of the text we use the notation: P-URAN[h, p], where h denotes the time (in hours) spent in the PLP stage (in addition to the first initialization time step); and p denotes the size of the partitions, i.e., partitions of p CPUs are formed from the overall N_c CPUs used in the computations (which means N_c/p number of partitions are used).

In this study we show that the P-URAN strategy yields the lowest wall clock time in all the cases tested; and shows good weak scaling up to 9216 cores.

3 LES/PDF simulation of Sandia Flame D

The aforementioned parallel strategies are tested by performing LES/PDF simulations of the Sandia Flame D.

3.1 Sandia Flame D

The Sandia Flame D is a piloted CH_4/Air jet flame operating at a jet Reynolds number, Re = 22,400. All the details about this flame and the burner geometry can be found at [26]. Here we mention only some of the important aspects of this flame.

The jet fluid consists of 25% CH_4 and 75% air by volume. The jet flows in at 49.6 m/s velocity at 294K temperature and 0.993 atm pressure. The jet diameter, $D = 7.2$ mm. The pilot is a lean (equivalence ratio, $\Phi = 0.77$) mixture of C_2H_2, H_2, air, CO_2, and N_2 with the same nominal enthalpy and equilibrium composition as that of CH_4/Air at this equivalence ratio. The pilot velocity is 11.4 m/s. The coflow is air flowing in at 0.9 m/s at 291K and 0.993 atm.

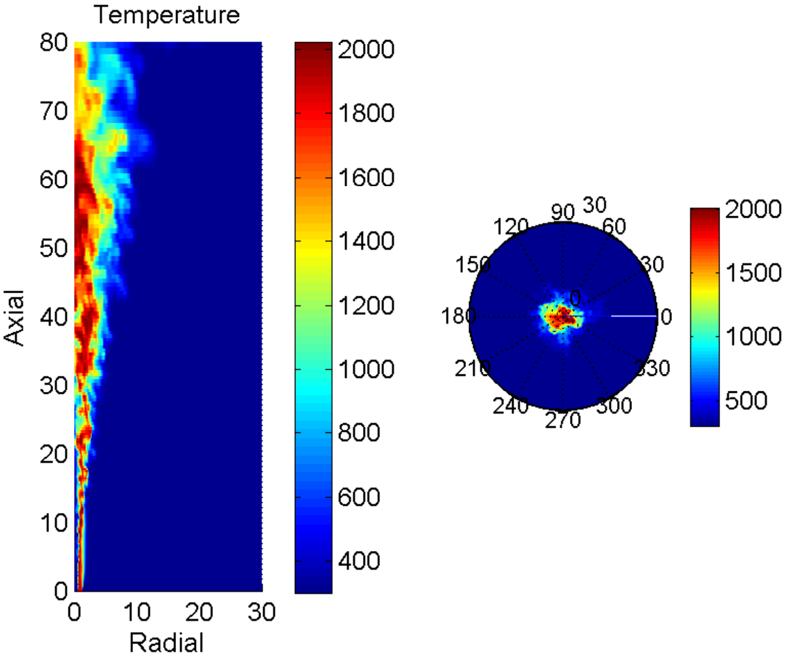

Figure 1: LES/PDF simulation of the Sandia Flame D. Instantaneous temperature distribution in the computational domain.

3.2 Computational Details

We perform LES/PDF simulation of the Sandia Flame D using the coupled LES/HPDF solver. The simulation is performed in cylindrical coordinate systems. A computational domain (see Fig.1) of $80\text{D}\times30\text{D}\times2\pi$ is used in the axial, radial and azimuthal directions, respectively. A non-uniform structured grid of size $192\times192\times96$ (in the axial, radial and azimuthal directions, respectively) is used for the LES solver. In the HPDF solver (for the base case), the number of particles per LES cell (N_{pc}) used is $N_{pc} = 40$. With a total of $192\times192\times96 \approx 3.5$M LES cells, an overall 141M particles are used in the computational domain. The chemistry is represented using the 16-species ARM1 reduced mechanism. A fixed ISAT error tolerance, $\epsilon_{tol} = 10^{-4}$ (which yields less than 1% tabulation error), is used in this study. All the simulations are performed on the Texas Advanced Computing Center (TACC) Ranger cluster.

The LES/PDF simulation tests are performed in three phases (more computational details are given in the results section later):

- Phase 1: Base Case

  In this phase we perform an LES/PDF simulation of the Sandia Flame D to obtain a statistically-stationary flame.

- Phase 2: Comparison of Parallel Strategies

  In this phase, starting from the statistically-stationary base case, we compare the performance of various parallel strategies implemented using *x2f_mpi*.

- Phase 3: Weak Scaling Studies

  In this phase we perform weak scaling studies with different *x2f_mpi* strategies using up to 9216 cores.

4 Results

In this section we present results for the previously mentioned three phases of testing performed.

4.1 Base Case

We perform an LES/PDF simulation of the Sandia Flame D on 1024 cores (using 64x16 domain decomposition) until a statistically-stationary state is reached. The LES/PDF time-averaged mean and variance statistics of the base case are found in good agreement with the experimental data (not presented here for brevity).

4.2 Comparison of Parallel Strategies

Starting from the base case, we employ the PLP, URAN and P-URAN parallel strategies, and compare the overall wall clock time for running 2000 simulation time steps. Here we consider P-URAN[0.2h, 32] strategy to test P-URAN; sensitivity results to changes in the time spent in the PLP stage and the partition size will be presented in the next section.

The timing results are shown in Fig.2 using bar charts. The bars also show the split-up of time spent in LES, HPDF (outside reaction), Waiting (synchronization time due to reaction load imbalance) and Reaction (including *x2f_mpi* communication time, if any). In the same figure, for comparison we also show the wall clock time if the chemistry is represented using a single scalar (mixture-fraction) based flamelet implementation. Also shown in the same figure are two estimates of the best wall clock time that can be achieved using ISAT/*x2f_mpi* if (i) all the CPUs have pre-built ISAT tables, and all the particles can be resolved by retrieving reaction mappings from the local tables; and (ii) the communication cost is zero, and chemistry workload is perfectly balanced among all the CPUs.

Based on these results we can draw the following conclusions:

1. the Waiting time, which is an indication of the extent of the load imbalance, is maximum for PLP, minimum for URAN (due to near-ideal load balancing), and moderate for P-URAN (mainly due to load imbalance across partitions);
2. in the P-URAN[0.2h, 32] strategy, more than 40% of the overall wall clock time is spent on reaction, confirming that reaction is the most expensive part of these computations;
3. the P-URAN[0.2h, 32] strategy takes 25% less wall clock time than PLP or URAN to perform 2000 simulation time steps;
4. the wall clock time with P-URAN[0.2h, 32] is within a factor of 1.5 of the best wall clock time estimate (based on no communication);
5. the wall clock time with P-URAN[0.2h, 32] for representing 16-species (ARM1 mechanism) chemistry is only 3 times as expensive as using the simple single-scalar based flamelet representation.

In short, we see that the P-URAN strategy performs much better than the PLP and URAN strategies, and is only within a factor of 1.5 of the best that can be achieved.

4.2.1 P-URAN: Sensitivity Tests

In the previous section we considered a specific P-URAN[0.2h,32] strategy for P-URAN, and so here we perform sensitivity studies to see how P-URAN performs with changes in the time spent in the PLP stage and the partition size.

First, we fix the partition size to 32, and vary the time spent in the PLP stage from 0 to 1 hour, and compute the overall wall clock time for running 2000 simulation time steps. The results are shown in Fig.3. We see that the P-URAN strategy shows very little sensitivity to changes in the time spent in the PLP stage. The reason for this is that when more time is spent in the PLP stage, the chances of retrieving from the local ISAT tables increases. So the initial time lost in the PLP stage due to load imbalance is compensated by local retrieves in the second stage of P-URAN. In general when performing long runs (for 24 hours or more), spending a good amount of time (say 1 hour) in the PLP stage should help build the local ISAT tables and increase the probability of local retrieves, thereby reducing the overall computation time.

Similarly, we fix the time spent in the PLP stage to 0.2h and change the partition size to 16, 32 and 64. The results are shown in Fig.4. Here again, the P-URAN strategy shows very little sensitivity to the changes in partition size. This presumably is because of a balance achieved between the communication cost and load imbalance. Smaller partitions reduce the communication cost but increase load imbalance between partitions, while bigger partitions increase communication cost but achieve better load balance. We suggest using a partition size which is a multiple of the domain decomposition in the radial (or lateral) direction, D_y, (and is preferably closer to $\sqrt{N_c}$) to strike a balance between the communication cost and load imbalance.

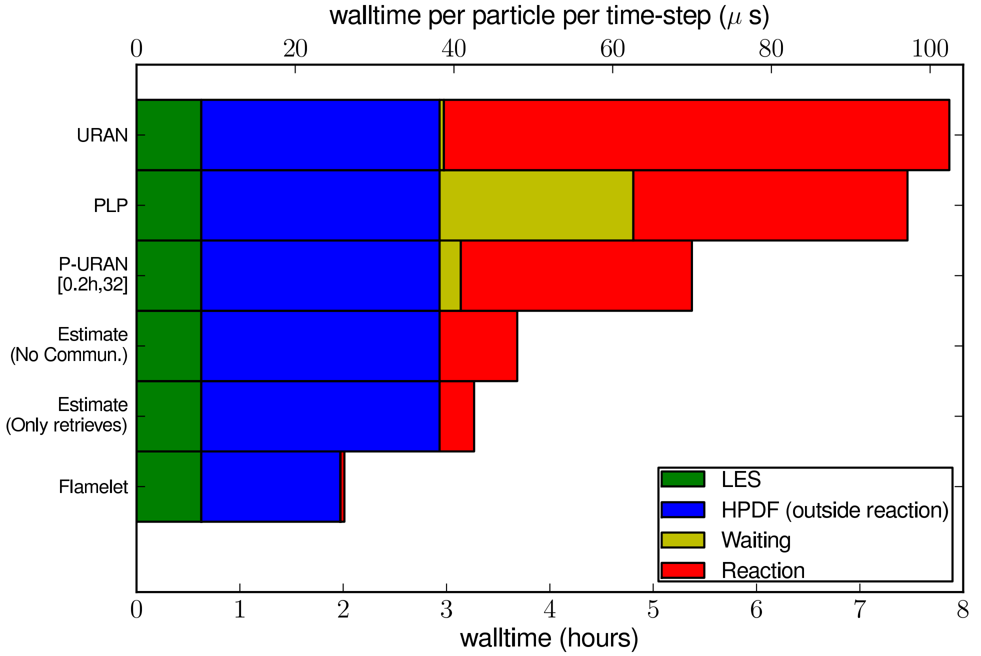

Figure 2: Wall clock time for 2000 time steps with different parallel strategies. From bottom to top: (1) Flamelet - using mixture-fraction based flamelet representation of chemistry; (2) Estimate (only retrieves) - estimate based on performing only local retrieves using pre-built ISAT tables; (3) Estimate (No Commun.) - estimate based on perfect load balancing with no communication cost; (4) PURAN[0.2h, 32]; (5) PLP and (6) URAN.

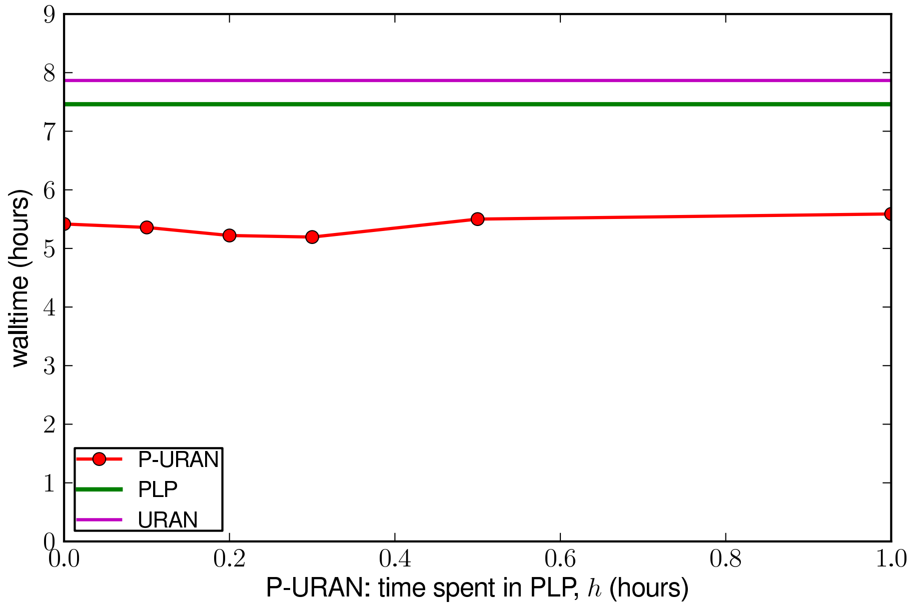

Figure 3: Wall clock time for 2000 time steps with (i) PLP; (ii) URAN; and (iii) P-URAN[h, 32]: with the time spent in the PLP stage, h, varied from 0 to 1 hour.

4.3 Weak Scaling Studies

We perform weak scaling tests with the URAN and P-URAN strategies on 2304 to 9216 cores. In these weak scaling tests, the HPDF workload per core is held constant by scaling the number of particles per LES cell appropriately. The details of the tests are listed in the Table 1. In each case, we perform an LES/PDF

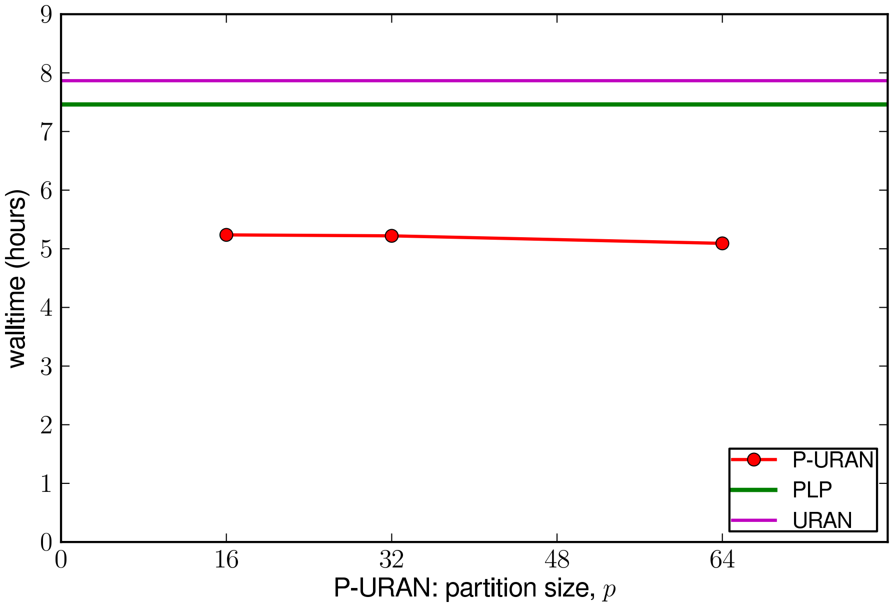

Figure 4: Wall clock time for 2000 time steps with (i) PLP; (ii) URAN; and (iii) P-URAN[0.2h, p]: with the partition size, $p = 16, 32, 64$.

simulation for 1000 time steps resulting in around 46M ISAT queries per core. The wall clock time spent for 1000 time steps using the URAN and P-URAN strategies is shown in Fig.5. We see that the P-URAN strategy consistently performs better than URAN on all the four core counts, and also shows better weak-scaling up to 9216 cores than the URAN strategy. Compared to results presented in Fig.2, in these weak scaling results the P-URAN strategy does not perform significantly better than URAN because these are relatively smaller runs (1000 time steps; due to limited availability of compute hours on the TACC Ranger cluster) and we expect the wall clock time with P-URAN to improve for longer runs due to increased probability of local retrieves and reduced communication cost.

Cores	Domain Decomposition	N_{pc}	Strategies
2304	48x48	30	URAN, P-URAN[0.2h, 48]
4608	96x48	60	URAN, P-URAN[0.2h, 48]
6144	96x64	80	URAN, P-URAN[0.2h, 64]
9216	96x96	120	URAN, P-URAN[0.2h, 96]

Table 1: Computational details of the weak-scaling tests performed using the URAN and P-URAN strategies.

5 Conclusions

We have successfully developed an integrated LES/HPDF/x2f_mpi/ISAT solver (scalable to 10,000 cores) for performing turbulent combustion calculations with real chemistry. We have demonstrated different parallel strategies (implemented using the *x2f_mpi* library) for performing chemistry calculations efficiently. In particular, we have developed a new parallel strategy, P-URAN, and shown that it

- yields the lowest wall clock time among all the strategies tested;
- is within a factor of 1.5 of an estimate for the lowest achievable wall clock time; and
- scales well up to 9216 cores.

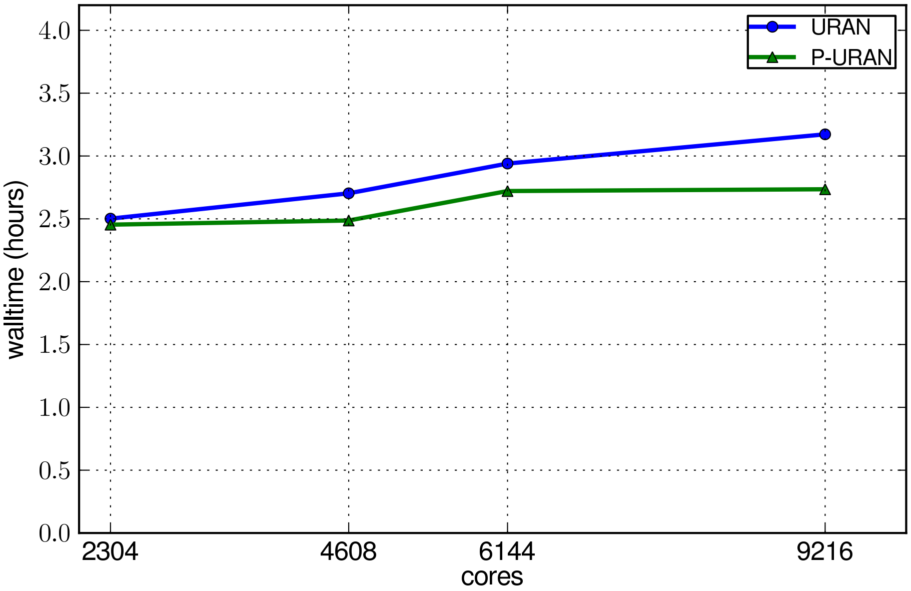

Figure 5: Weak scaling tests: wall clock time with URAN and P-URAN strategies.

6 Acknowledgments

This work is supported in part by Department of Energy under Grant DE-FG02-90ER and by Grant Number FA9550-09-1-0611 funded by the National Center for Hypersonic Combined Cycle Propulsion, sponsored by the AFOSR and NASA ARMD. This research was also supported in part by the National Science Foundation through TeraGrid resources provided by the Texas Advanced Computing Center under Grant No. TG-CTS090020. SBP has a financial interest in Ithaca Combustion Enterprise, LLC., which has licensed the software ISAT-CK and CEQ used in this work.

References

[1] W.J. Pitz, N.P. Cernansky, F.L. Dryer, F.N. Egolfopoulos, J.T. Farrell, D.G. Friend, and H. Pitsch, *Development of an Experimental Database and Kinetic Models for Surrogate Gasoline Fuels.*, Society of Automotive Engineers SAE-2007-01-0175 (2007).

[2] C.K. Westbrook, W.J. Pitz, O. Herbinet, H.J. Curran, and E.J. Silke, *A comprehensive detailed chemical kinetic reaction mechanism for combustion of n-alkane hydrocarbons from n-octane to n-hexadecane.*, Combust. Flame 156 (2009), pp. 181–199.

[3] T. Lu and C.K. Law, *A directed relation graph method for mechanism reduction.*, Proc. Combust. Inst. 30 (2005), pp. 1333–1341.

[4] P. Pepiot-Desjardins and H. Pitsch, *An efficient error-propagation-based reduction method for large chemical kinetic mechanisms.*, Combustion and Flame 154 (2008), pp. 67–81.

[5] T. Nagy and T. Turanyi, *Reduction of very large reaction mechanisms using methods based on simulation error minimization*, Combustion and Flame 156 (2009), pp. 417–428.

[6] J.C. Keck and D. Gillespie, *Rate-controlled partial equilibrium method for treating reacting gas-mixtures.*, Combust. Flame 17 (1971), pp. 237–241.

[7] S.H. Lam and D.A. Goussis, *The CSP Method for Simplifying Kinetics.*, International Journal of Chemical Kinetics 26 (1994), pp. 461–486.

[8] U. Maas and S.B. Pope, *Simplifying chemical-kinetics: intrinsic low-dimensional manifolds in composition space.*, Combust. Flame 88 (1992), pp. 239–264.

[9] Z. Ren, S.B. Pope, A. Vladimirsky, and J.M. Guckenheimer, *The invariant constrained equilibrium edge preimage curve method for the dimension reduction of chemical kinetics.*, J. Chem. Phys. 124 (2006), p. Art. No. 114111.

[10] L. Lu and S.B. Pope, *An improved algorithm for in situ adaptive tabulation.*, Journal of Computational Physics 228(2) (2009), pp. 361–386.

[11] S.R. Tonse, N.W. Moriarity, N.J. Brown, and M. Frenklach, *PRISM: Piecewise Reusable Implementation of Solution Mapping. An economical strategy for chemical kinetics.*, Israel J. Chem. 39 (1999), pp. 97–106.

[12] T. Turanyi, *Parameterization of reaction mechanisms using orthonormal polynomials.*, Computers Chem. 18 (1994), pp. 45–54.

[13] V. Hiremath, Z. Ren, and S.B. Pope, *A Greedy Algorithm for Species Selection in Dimension Reduction of Combustion Chemistry*, Combustion Theory and Modelling 14(5) (2010), pp. 619–652.

[14] V. Hiremath, Z. Ren, and S.B. Pope, *Combined Dimension Reduction and Tabulation Strategy using ISAT-RCCE-GALI for the Efficient Implementation of Combustion Chemistry*, Combustion and Flame (2011), p. doi:10.1016/j.combustflame.2011.04.010.

[15] L. Lu, Z. Ren, V. Raman, S.B. Pope, and H. Pitsch, *LES/FDF/ISAT computations of turbulent flames*, Proceedings of the CTR Summer Program (2004), pp. 283–294.

[16] L. Lu, Z. Ren, S.R. Lantz, V. Raman, S.B. Pope, and H. Pitsch, *Investigation of strategies for the parallel implementation of ISAT in LES/FDF/ISAT computations*, Proceedings of the 4th Joint Meeting of US Sections of the Combustion Institute, Philadelphia, PA (2005).

[17] L. Lu, S.R. Lantz, Z. Ren, and S.B. Pope, *Computationally efficient implementation of combustion chemistry in parallel PDF calculations*, Journal of Computational Physics 228(15) (2009), pp. 5490–5525.

[18] Y. Shi, W.H. Green, H.W. Wong, and O.O. Oluwole, *Redesigning combustion modeling algorithms for the Graphics Processing Unit (GPU): Chemical kinetic rate evaluation and ordinary differential equation integration*, Combustion and Flame 158(5) (2011), pp. 836–847.

[19] C.D. Pierce, *Progress-Variable Approach for LargeEddy Simulation of Turbulent Combustion*, Stanford University, 2001.

[20] C.D. Pierce and P. Moin, *Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion*, Journal of Fluid Mechanics 504 (2004), p. 7397.

[21] H. Wang and S.B. Pope, *Large Eddy Simulation/Probability Density Function Modeling of a Turbulent CH4/H2/N2 Jet Flame*, Proceedings of the Combustion Institute 33 (2011), pp. 1319–1330.

[22] H. Wang, P.P. Popov, and S.B. Pope, *Weak Second Order Splitting Schemes for Lagrangian Monte Carlo Particle Methods for the Composition PDF/FDF Transport Equations*, Journal of Computational Physics 229 (2010), pp. 1852–1878.

[23] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Dierential Equations*, Springer Verlag Berlin (1992).

[24] J. Janicka, W. Kolbe, and W. Kollmann, *Closure of the transport equation for the PDF of turbulent scalar fields*, Journal of Non-Equilibrium Thermodynamics 4 (1970), pp. 47–66.

[25] M. Caracotsios and W.E. Stewart, *Sensitivity analysis of initial value problems with mixed ODEs and algebraic equations.*, Computers & Chemical Engineering 9 (1985), pp. 359–365.

[26] R.S. Barlow and J.H. Frank, *Sandia/TUD Piloted CH4/Air Jet Flames*, http://www.sandia.gov/TNF/DataArch/FlameD.html .