Shared Memory Parallelism for 3D Cartesian Discrete Ordinates Solver

Salli Moustafa, Ivan Dutka Malen, Laurent Plagne, Angélique Ponçot, Pierre Ramet

Shared Memory Parallelism for 3D Cartesian Discrete Ordinates Solver

Salli Moustafa[1], Ivan Dutka-Malen[1], Laurent Plagne*[1], Angélique Ponçot[1], and Pierre Ramet[2]

[1]*EDF R&D 1, Av du Général de Gaulle F92141 CLAMART CEDEX France*
[2]*INRIA-University of Bordeaux, France*

This paper describes the performance of DOMINO, a 3D Cartesian SN Solver that implements two nested levels of parallelism (multicore+SIMD) on shared memory computation nodes. DOMINO solves a 3D full core PWR k_{eff} problem involving 26 groups, 288 angular directions (S16), 46×10^6 spatial cells and 1×10^{12} DoFs within 12 hours on a 32 core SMP node. This represents a sustained performance of 184 GFlops and 31% of the SMP node peak perfomance.

KEYWORDS: Deterministic transport, SN, multicore processor, wavefront, TBB, SIMD

I. The DOMINO COCAGNE SN Solver

COCAGNE is the name of the new nuclear core simulation system developed at EDF R&D. Recently, a shared-memory parallel solver named DOMINO based on a discrete ordinates method has been introduced into the COCAGNE system.

1. DOMINO Numerical Schemes

DOMINO implements the Discrete Ordinates Method for the stationary neutron transport in a multi-dimensional Cartesian geometry. The quadrature used is the level symmetric one. In three-dimensions, this quadrature leads to $N(N+2)$ discrete directions where N stands for the method's order. The space discretization corresponds to the Diamond Differencing scheme DD0. The DD0 element has 1 moment and 3 mesh-edge incoming fluxes per cell. The spatial problem is solved through the well-known sweep algorithm. We use a classical Diffusion Synthetic Acceleration (DSA) method.

2. Shared Memory Parallel Implementation of DOMINO

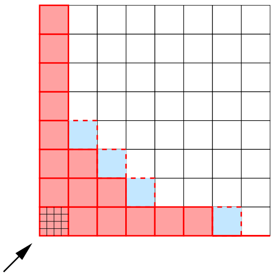

Figure 1: Wave front algorithm applied to the sweep across the mesh macrocells. The incoming fluxes come from the bottom left corner. The treated macrocells are pink colored while the white macrocells are still waiting. Only blue dashed border macrocells are ready for treatment and their respective indexes are stored in a task list from which a set of threads concurrently pick their work item. This list is dynamically updated each time a new macrocell has been treated.

The sweep operation is the most computationally intensive portion of DOMINO. For each incoming direction of the angular quadrature, the angular and volumic flux Degrees of Freedom (DoFs) of each spatial mesh cell must be updated. For simplicity's sake, we illustrate the parallel sweep on a 2D example with a DD0 spatial discretization scheme. Let $\{c_{ij}\}$ be the $n_x \times n_y$ cells of a 2D Cartesian spatial mesh. Let us consider Ω_d an angular direction incoming from the bottom left corner. Let $\{{}^d\psi^x_{ij}, {}^d\psi^y_{ij}, \phi_{ij}\}$ be the corresponding incoming angular and volumic flux DoFs. In the sweep operation one has to treat each cell c_{ij} by updating the volumic ϕ_{ij} and the angular outcoming ${}^d\psi^x_{i+1j}$ and ${}^d\psi^y_{ij+1}$ DoFs that depend on the three ${}^d\psi^x_{ij}, {}^d\psi^y_{ij}$ and ϕ_{ij} input values. This implies an ordering constraint for the sweep operation: a cell c_{ij} can only be treated if the following two conditions are fulfilled:

$$\begin{aligned} &c_{i-1,j} \text{ is already treated or } i = 0,\\ &c_{i,j-1} \text{ is already treated or } j = 0. \end{aligned}$$

Obviously c_{00} is the first cell that can be treated but the second can be either c_{10} or $c_{01}\ldots$ or both can be treated in parallel. We rely on the Intel TBB primitive `parallel_do`$^{(1)}$ that enables a dynamic scheduling of the parallel tasks to implement the sweep. This parallel function allows a set of threads to execute the tasks from a task list which is dynamically updated. In the beginning, the task list contains the cell c_{00}. One of the running threads treats this cell and updates the task list to $\{c_{10}, c_{01}\}$ and so on. In order to reduce the overhead due to the thread scheduling, the cells are not treated individually but they are packed into groups of cells, called macrocells in the DOMINO implementation. All the cores of the supercomputing nodes are used in this parallel wavefront algorithm. Figure 1 illustrates this parallel sweep strategy.

During each sweep, the angular directions that belong to the same octant are computed via SIMD instructions allowing to perform simultaneously two double precision or four simple precision floating point operations.

In this paper the computations are made with one of the computing nodes of the EDF R&D Ivanoé supercomputer[1]. This 1024 GB node is a 4-socket Intel X7560 processor (2.26GHz), 8 cores per socket, and 4-way SIMD units totalizing 128 parallel FPUs that can be used concurrently. As a consequence, one can expect a 128 fold speed-up for an ideal parallel implementation of DOMINO compared to a sequential non-SIMD run time.

II. Scalability and Efficiency

We analyze the performance of the sweep operation for a simple 2-group k_{eff} computation with 480^3 spatial cells with different angular quadrature orders. Figure 2 and Table 1 sumarize the performance of the 2-level parallel implementation (multicore+SIMD) of DOMINO. Since the arithmetic intensity of the computation increases with the direction number, the sweep performance improves with the angular quadrature order. S2 and S4 timings are exactly the same because we use SIMD units to treat 4 angular directions that belong to the same octant at the same time. GFlops value is estimated by dividing the floating point operation number by the completion time:

$$\text{GFlops} = \frac{25 \times N_{\text{cells}} \times N_{\text{dir}}}{\text{Time in nanosecond}},$$

where the factor 25 accounts for the 20 add/mult operations and 1 division per cell (we count 5 flops for the division).

In the S16 case ($N_{\text{dir}} = 288$), the sweep performance reaches 216 GFlops which corresponds to 37% of the peak performance given by (Single Precision):

$$\begin{aligned} \text{Peak} &= \text{ncores} \times \text{frequency} \times \text{SIMD width} \times \text{SIMD units} \\ &= 32 \times 2.26 \times 4 \times 2 = 578 \text{ GFlops}. \end{aligned}$$

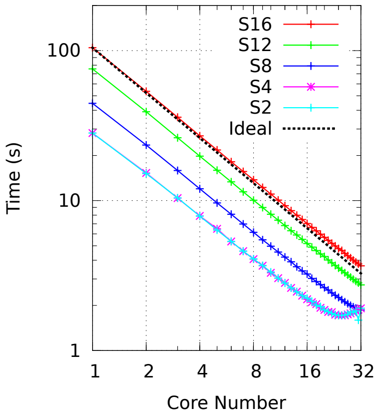

Figure 2: **Time per sweep over all angular directions as a function of the core number (mesh: 480 × 480 × 480).**

III. DOMINO/MCNP Comparison

The benchmark used for calculations is described in reference.[(2)] It corresponds to a simplified 3D PWR first core loaded with 3 different types of fuel assemblies characterized by a specific Uranium-235 enrichment (low, medium and highly enriched uranium).

Case	N_{dir}	Seq. Time (s)	32-Core Time (s)	Speed Up	GFlops	% Peak Perf.
S2	8 × 1	28.2	1.92	14.7	11.5	2%
S4	8 × 3	28.2	1.92	14.7	34.7	6%
S8	8 × 10	44.5	1.86	23.9	118.7	20.5%
S12	8 × 21	75.6	2.74	27.6	169.6	29.3%
S16	8 × 36	104.7	3.67	28.5	216.8	37.5%

Table 1: Time per sweep over all angular directions as a function of the core number (mesh: 480 × 480 × 480).

The calculation mesh is based on the cell pin mesh (289×289) and along the z-axis, the height is subdived into 70 meshes for the 26-group calculation. For spatially converged results, meshes are refined by 2 in directions x, y and z. The angular discretization corresponds to a S16 level symmetric angular quadrature, which results in 288 directions in 3D. To simplify the comparisons to MCNP results, the 3D 26-group fluxes are integrated over energy and space. Practically, a group collapsing is performed from 26 to 2 energy groups with a boundary fixed at 0.625 eV, and a spatial integration over each pin-cell along the z-axis is made. Comparisons between DOMINO and MCNP presented in Table 2 are computed in the form:

$$\Delta k_{eff} = \frac{k_{eff}^{DOMINO} - k_{eff}^{MCNP}}{k_{eff}^{MCNP}} \times 10^5 \text{ in pcm},$$

$$| \delta\phi_g |= \max_{\{p_i\}} | \frac{\phi_g^{DOMINO}(p_i) - \phi_g^{MCNP}(p_i)}{\phi_g^{MCNP}(p_i)} |,$$

where p_i stands for the pin cell index.

k_{eff}	Δk_{eff} (pcm)	$\mid \delta\phi_1 \mid$ (%)	$\mid \delta\phi_2 \mid$ (%)	$\mid \delta\nu\Sigma_f\phi \mid$ (%)	DOMINO time
1.008361	12	0.69	0.80	0.34	12h06

Table 2: Comparison to MCNP/DOMINO: S16, 26-group, 578 × 578 × 140 spatial cells. 2.26GHz SMP node with 32 cores

This SN k_{eff} calculation involved 1.05×10^{12} DoFs obtained within 126 power iterations for a 10^{-5} stopping criterium. The DOMINO implementation reached 184.3 GFlops which corresponds to 31% of the SMP node peak performance.

Encouraged by these node performances, we are currently working on extending DOMINO to distributed memory machines.

References

1) A. Robison, M. Voss, and A. Kukanov, "Optimization via reflection on work stealing in TBB," *Proc. Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, p. 1–8, IEEE, 2008.

2) T. Courau, L. Plagne, A. Ponçot, and G. Sjoden, "Hybrid Parallel Code Acceleration Methods in Full-Core Reactor Physics Calculations," *Proc. Physor 2012 - Advances in Reactor Physics*, 2010.

[1] Ranked 149 in Nov 2012 TOP 500 List