UMT

Summary Version
1.2

Purpose of Benchmark

UMT is an LLNL ASC proxy application (mini-app) that performs three-dimensional, non-linear, radiation transport calculations using deterministic (Sn) methods.

Characteristics of Benchmark

The method of solution is as follows. UMT performs the solution of time-dependent, energy-dependent, discrete ordinates, and nonlinear radiation problems on three-dimensional unstructured spatial grids on large distributed-memory multi-node parallel computer systems with multiple cores per node. To achieve extreme scalability, the application exploits both spatial decomposition using message passing between nodes and a threading algorithm in angle within the node.

The project addresses the need to obtain accurate transport solutions to three-dimensional radiation transfer problems (i.e. the transfer of thermal photons). The solution calls for the use of an unstructured grid.This class of problems is characterized by tens of thousands of unknowns per zone and upwards of millions of zones, thus requiring large, scalable, parallel computing platforms. The package uses a combination of message passing and threading, utilizes the large distributed memory of the platform and provides unprecedented (weak) scaling to very large core counts.

UMT requires a minimum of 1GB of main memory per MPI task.

Mechanics of Building Benchmark

1. cd `UMT2013`
2. Modify `make.defs` to reflect the platform's compilers, compiler options, libraries, MPI wrappers etc. A working/tested example is provided.
3. `gmake clean`
4. `gmake`
5. The following libraries or their ".a" counterparts should exist at this point:

```
./CMG_CLEAN/src/libcmgp.so
./Teton/libTetonUtils.so
./Teton/libInfrastructure.so
./cmg2Kull/sources/libc2k.so
```

Note:

The code is designed to use MPI and OpenMP.
SIMD and/or other vectorization may be turned on and is encouraged.

6. `cd ./Teton` (now in `UMT2013/Teton`)
   `gmake SuOlsonTest`
7. The executable is: `/ UMT2013/Teton/SuOlsonTest`

Mechanics of Running Benchmark

Example command line parameters or inputs:
For all cases below, set these values:
Order=16; Groups=16; quadType=2; Polar=8; Azim=4
In addition, the input parameter $gridFileName is the name of the input file. Several examples of input files are provided.

1. Small problem: single node and/or single CPU
   gridFileName=problem1.cmg # this file is included.
   export OMP_NUM_THREADS=1
   *srun –n 1 –N 1 ./SuOlsonTest $gridFileName *
   $Groups $quadType $Order $Polar $Azim
2. Medium problem: (<1K node job)
   gridFileName= grid_8192_12x12x12.cmg # this file is included
   export OMP_NUM_THREADS=8
   *srun –n 8192 –N 1024 ./SuOlsonTest $gridFileName *
   $Groups $quadType $Order $Polar $Azim
3. Large CORAL Reference (used for reference (FOM measurement on IBM BG/Q baseline) problem:
   Throughput: 4K node job on Sequoia
   gridFileName= grid_32768_12x12x12.cmg # this file is included
   export OMP_NUM_THREADS=8
   *srun -n 32768 -N 4096 ./SuOlsonTest $gridFileName *
   $Groups $quadType $Order $Polar $Azim
4. CORAL class problem (2X the size of CORAL reference problem):
   gridFileName= grid_65536_12x12x12.cmg # this file is included
   export OMP_NUM_THREADS=32
   *srun -n 65536-N 8192 ./SuOlsonTest $gridFileName *
   $Groups $quadType $Order $Polar $Azim

Details are provided in the README included in the UMT2 tarfile.

Verification of Results

How are the benchmark results verified for correct answers?

A typical output file is provided in the Teton directory: Intel_SNB_64MPIx8omp_32nodes.out
This file was the output of an execution of SuOlsonTest with 64 MPI tasks on 32 nodes (16cores/node) and 8 OMP threads per rank.

What FOM should be reported?

UMT2 computes FOM and reports it. FOM is computed as follows:

*FOM = number_Unknowns/cumulativeWorkTime*cumulativeIterationCount.*
For weak scaling this FOM grows almost linearly as more resources (ranks) are added. Each MPI rank processes the same number of zones set with the numzones(Nx, Ny, Nz) specification in the input file. The FOM measures the total amount of work done by the entire job.