Trinity Capability Improvement Metric

Introduction

The purpose of this document is to provide a detailed description and run rules for the applications to be used for the Trinity Capability Improvement (CI) metric. There will be three (3) applications used, one from each of the three NNSA laboratories: SIERRA (SNL), PARTISN (LANL) and Qbox (LLNL). These applications were selected as representative workloads and contain the types of calculations that Trinity needs to be capable of in support of the NNSA's mission.

The CI metric is defined as the product of an increase in problem size, and/or complexity, and an application specific runtime speedup factor. For example, if the problem size increases by a factor of 8 and the runtime speedup is 1.2, the CI is 8x1.2=9.6. Please see the Trinity RFP for a complete definition of the Capability Improvement requirement.

For each application, a baseline result has been obtained on the NNSA's Cielo platform [1]. All CI factors will be calculated based on the Cielo results. Note that it is not a requirement to provide a CI factor for each of the three applications as a part of a proposal, it is required that you agree to meet, on average, the minimum CI specified in the requirements document, at time of acceptance.

Application Descriptions

SIERRA:

Description: The SIERRA Low Mach Module (internal code name: Nalu) solves a wide variety of variable density acoustically incompressible flows spanning from laminar to turbulent flow regimes. This generalized unstructured code base supports both elemental (control volume finite element) and edge (edge-based vertex centered) discretizations in the context of an approximate pressure projection algorithm (equal order interpolation using residual-based pressure stabilization). The generalized unstructured algorithm is second order accurate in space and time. A variety of turbulence models are supported, however, all are classified under the class of modeling know as Large Eddy Simulation (LES). The chosen coupling approach (pressure projection, operator split) results in a set of fully implicit sparse matrix systems. Linear solves are supported by the Trilinos Tpetra interface. Finally, this multi-physics simulation tool is built under the SIERRA Toolkit.

By deploying a code base written to leverage both Trilinos/Tpetra and the SIERRA Toolkit, i.e., codes bases that have been demonstrated to be 64-bit compliant and represent the path towards advanced architectures; the Nalu simulation tool can

support mesh and degree-of-freedom counts well above the 2.14 billion count. The calculations are computationally intensive and require good cache usage. In typical applications, hundreds of thousands of time steps must be used. Communication patterns include both point-to-point exchanges typical of sparse graphs, consistent with assembly of partial sums, and collective reduction operations including global minimums, maximums, and summations. This code base is fairly representative of a wide range of implicit codes that have been developed in support of the Advanced Simulation and Computing (ASC) Integrated Codes (IC) project.

Problem Description: The test problem of interest is a turbulent open jet (Reynolds number of ~6,000) with passive mixture fraction transport using the one-equation Ksgs LES model. The problem is discretized on unstructured meshes with hexahedral elements. The baseline problem mesh consists of nine billion elements, with total degree-of-freedom count approaching 60 billion. Given the pressure projection scheme in the context of a monolithic momentum solve, the maximum matrix size is ~27 billion rows (momentum) followed by a series of smaller 9 billion row systems, i.e., for the continuity system (elliptic Pressure Poisson), mixture fraction and turbulent kinetic energy. For the acceptance criteria, the mesh will be uniformly refined one level to increase the size of the problem by a factor of eight.

Figure of Merit Description: Two figures of merit will be employed; both involve the solution of the momentum equations. The speedup of the two metrics will be weighted to produce a single speedup factor for SIERRA. The first figure of merit will be the average "solve" time per linear iteration. The second will be the average matrix "assemble" time per nonlinear step. Speedup will be defined as: Speedup = $\text{Speedup}_{\text{solve}}*0.67 + \text{Speedup}_{\text{assemble}}*0.33$.

Point of contact for distribution: Mike Glass, Sandia National Laboratories, mwglass@sandia.gov, (505) 844-8451.

PARTISN

Description: PARTISN, "An Sn Algorithm for the Massively Parallel CM-200 Computer", Randal S. Baker and Kenneth R. Koch, Nucl. Sci. and Eng., Vol. 128, pp. 313-320 (1998), provides neutron transport solutions on orthogonal meshes in one, two, and three dimensions. A multigroup energy treatment is used in conjunction with the Sn angular approximation. Much effort has been devoted to making PARTISN efficient on massively parallel computers. The package can be coupled to nonlinear multiphysics codes that run for weeks on thousands of processors to finish one simulation.

The primary components of the computation involve KBA sweeps and associated zero-dimensional physics. The KBA sweep is a wavefront algorithm that provides 2-D parallelism for 3-D geometries, and is tightly coupled by dependent communications. In general, PARTISN is more sensitive to communication latency and processor speed then memory bandwidth. PARTISN relies heavily on

MPI_Send/Recv, the most frequent collective is MPI_AllReduce. For a 1024 rank run the code executed around 6M sends, 6M recvs, and approximately 4K allreduces.

Problem Description: The problem provided is "sntiming". The problem is to be weak-scaled, trying to keep an evenly cubed space x*y*z (i.e. where x,y and z are approximately the same). As some of the parameters are non-intuitive to derive, we will provide a scaling spreadsheet to show input parameters for increasing numbers of ranks.

Figure of Merit Description: For PARTISN we will record "Transport Grind Time", i.e., the time to perform one iteration over one phase space cell. Ideally, the Transport Grind Time should decrease linearly as you scale out in PE's.

Point of contact for distribution: Manuel Vigil, Los Alamos National Laboratory, mbv@lanl.gov, (505) 665-1960.

Qbox

Description: Qbox is a first-principles molecular dynamics code used to compute the properties of materials at the atomistic scale. The main algorithm uses a Born-Oppenheimer description of atomic cores and electrons, with valence electrons treated quantum mechanically using Density Functional Theory and a plane wave basis. Nonlocal pseudopotentials are used to describe the core electrons and nuclei, and derived to match all-electron single atom calculations outside of a given cutoff radius. The computational profile consists primarily of parallel dense linear algebra and parallel 3D complex-to-complex Fast Fourier Transforms. Efficient single-node kernels have been found to be necessary to achieve good peak performance. The communication patterns are complex, with nonlocal communication occurring both within the parallel linear algebra library (ScaLAPACK) and in subcommunicator collectives within Qbox, primarily MPI_Allreduce and MPI_Alltoallv operations. Threading is currently implemented as a mix of OpenMP and threaded single-node linear algebra kernels supplied by the hardware vendor.

Problem Description: The Qbox benchmark problem is the initial self-consistent wavefunction convergence of a large crystalline gold system (fcc, a_0 = 7.71 a.u). This problem is computationally identical to typical capability simulations of high-Z materials, but easier to describe and generalize to arbitrary numbers of atoms. A script is provided to generate input files of any sized system (examples/gold_benchmark/qbox_gold_makeinputs.pl). On Cielo, N=1600 gold atoms were simulated with a norm-conserving pseudopotential and 17 valence electrons per atom, resulting in 13,600 occupied electronic orbitals. A planewave cutoff of 130 Rydbergs was used, and 2784 additional unoccupied orbitals included (approximately 20% of the number of occupied states) to allow for finite temperature smearing of the occupation at the Fermi level. The computational complexity of the calculation scales as $O(N^3)$ where N is the number of electronic orbitals, and the number of atoms should be increased accordingly to generate

inputs for Trinity. For example, 2880 gold atoms would be approximately six times more work than the Cielo benchmark system of 1600 atoms.

Figure of Merit Description: The run time metric for this benchmark is the maximum total wall time to run a single self-consistent iteration with three non-self-consistent inner iterations (corresponding to an input command of "run 0 1 3"). Qbox prints formatted XML tags for the timing of each part of the code at the end of the run, with the self-consistent iteration time marked as follows:

<timing where="run" name=" iteration" min="1234.5 " max="1234.5 "/>

The figure of merit is the timing in the max field.

Point of contact for distribution: Erik Draeger, Lawrence Livermore National Laboratory, draeger1@llnl.gov, (925) 423-3650.

Table 1: ASC Simulation Code Suite Summary

	Languages				Current Parallel APIs		
Application	Fortran	C	C++	Python	MPI	OpenMP	Description
SIERRA	X		X		X		Variable density acoustically incompressible fluid flows on unstructured meshes; Large Eddy Simulation; solution of implicit sparse matrices.
PARTISN	X				X	X	Neutron transport with adaptive mesh refinement
Qbox			X		X	X	Molecular dynamics at the atomic scale, quantum mechanics using density functional theory and a plane wave basis

Cielo Results

Table 2: Cielo Baseline Performance Data

	# Nodes	# MPI ranks	Problem Size/Complexity	Figure of Merit
SIERRA	8,192	131,072	9e9 elements, 27e9 row matrix	Solve time/linear iteration = 0.521 seconds; Matrix Assemble time/nonlinear step = 2.42 seconds
PARTISN	8,192	131,072	226,981,000 cells, 30 energy groups, 16 angular quadrature order	Transport Grind Time = 3.3 picoseconds per cell
Qbox	6,144	98,304	1,600 gold atoms	Iteration time = 1,663 seconds

References

[1] http://www.lanl.gov/orgs/hpc/cielo/index.shtml/, NNSA/ASC Cielo Supercomputer.

[2] "A Block Adaptive Mesh Refinement Algorithm for the Neutral Particle Transport Equation", Randal S. Baker, Nuclear Science and Engineering, 2002-05-15, vol 14, no 1, pg 1--12, issn 0029-5639