PARTISN: Enabling a 20th Century FORTRAN Transport Code to Meet 21st Century Challenges

M&C Roundtable

ANS Annual Meeting, 2021

Erin J. Davis, CCS-2

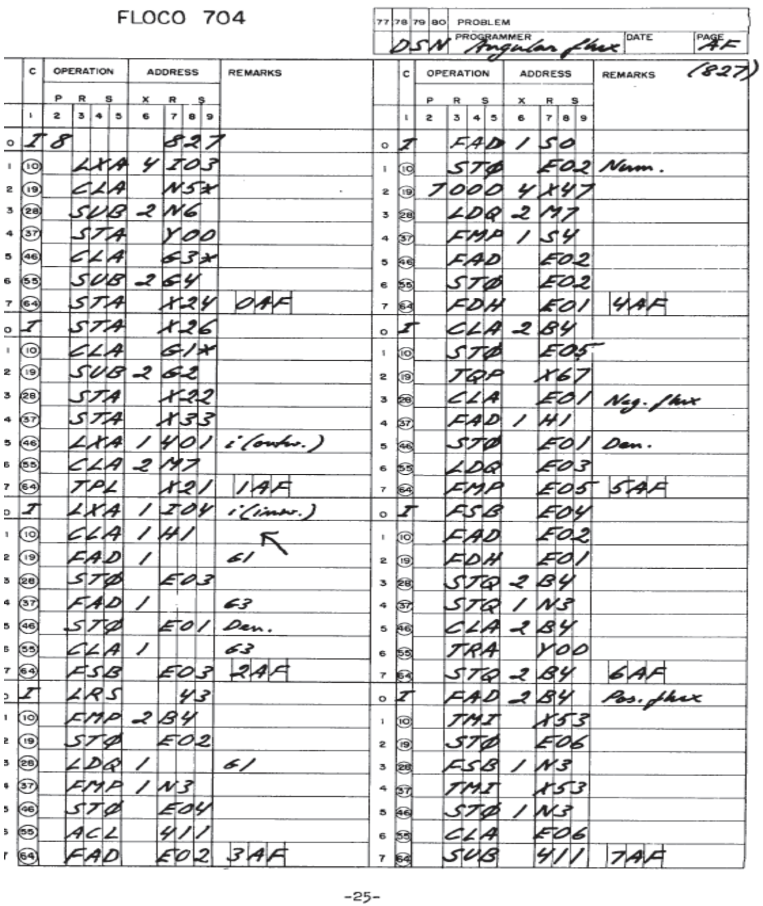
June 14, 2020

Los Alamos
NATIONAL LABORATORY
EST. 1943

NNSA
National Nuclear Security Administration

Ancient History

- DSN (1-D) and TDC (2-D) codes developed at LANL in the late 1950s
  - Authors: B. Carlson, C. Lee and J. Worlton
  - IBM 704 Calculator (12k FLOPs/sec, 8K 36-bit words)
  - Written in Floco II
    - DSN – 1800 words
    - TDC – 2000 words
- LANL's 1-D/2-D production S_N transport capability through the early 1960s



Modified from slides by Randy Baker

Ancient History

- DTF IV (1965)
  - Authors: K. Lathrop
  - 1-D Cartesian and curvilinear geometries
  - Written in Fortran IV
    - 15 subroutines (8000 words)
- ONETRAN, TWOTRAN, TWOHEX, etc. (1960s)
  - Authors: K. Lathrop, D. O'Dell, F. W. Brinkley, D. Marr, W. Walters
  - "Modern" form of PARTISN

CDC-7600

CDC-6600

- 1st successful supercomputer (Seymour Cray)
- 60-bit words
- 10 MHz RISC-based CPU
- 256K words central memory
- Max program length of 128K words

Modified from slides by Randy Baker

Ancient History

- ONEDANT, TWODANT, THREEDANT, DANTSYS
  - Authors: R. Alcouffe, D. O'Dell, F. W. Brinkley, D. Marr and W. Walters
  - DANTSYS (1988) was ~500k lines of code

1976 – Cray 1

1982 – Cray X-MP

1988 – Cray Y-MP

- DANTSYS ported to the CM-200
  - CM Fortran
  - Massively parallel sweeps

Early 1990s – CM-200

First large scale, massively parallel machines at LANL

- DANTSYS/T3D
  - MPI

1993-1995 – Cray T3D

- S_N transport code development at LANL has occurred in lockstep with evolving computing resources

Modified from slides by Randy Baker

PARTISN : PARallel, TIme-dependent SN

- Coupled neutron-gamma deterministic transport: S_N + multigroup
- Geometries: 1-D (slab, two-angle slab, cylindrical, and spherical), 2D (x-y, r-z, and r-theta) and 3-D (x-y-z and r-z-theta)
- Direct (forward) and adjoint
- **Stochastic neutronics (POI, moments of fission and neutron #)**
- Inhomogeneous (fixed) source, k- and α-eigenvalues, nuclide concentration, dimensional search
- Time-dependent calculations
- Boundary conditions: Vacuum, reflective, periodic, white
- Arbitrary anisotropic scattering order (P_N, Galerkin scattering)
- Strictly positive scattering source option
- Group- and time-dependent quadrature
- Diamond-difference (DD), adaptive weighted DD, linear discontinuous, exponential discontinuous spatial differencing
- Automatic mesh coarsening
- **Block Adaptive Mesh Refinement (AMR)**
- Nonlinear diffusion acceleration
- **Self-shielding corrections for multigroup cross sections**
- Vectorized transport sweep
- **Parallel processing (CPU, MIC, ARM, GPU) via MPI+X (OpenMP, CUDA Fortran)**

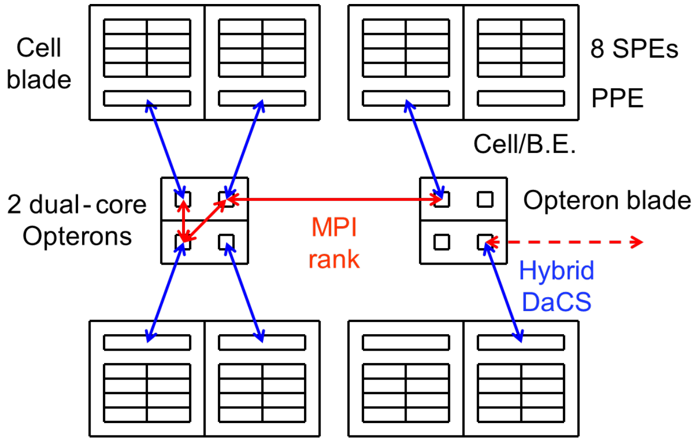
- **PARTISN came into existence in the late 1990s under ASC program**
  – Renamed DANTSYS/T3D
  – Large parts rewritten in Fortran 90/95
  – Massively parallel platforms
- Today, ~115k lines of (mostly) modern Fortran + CUDA Fortran
- There are 10 developer names on the PARTISN manual, but many others have contributed
  – *The code base represents a large body of transport knowledge and application experience going back to the 1950s*
- PARTISN has a lot of capability—Time-consuming to replicate and V&V
- Rewriting in C/C++/??? has never been off the table, *but* we have thus far been very successful at porting to new architectures

“Mature” Fortran Code Meets Advanced Architectures

- The initial PARTISN port to Roadrunner was completed in 2011
  - 3 developers working half-time for 2 years
    - Randy Baker, Chuck Ferenbaugh & Brian Lally
    - Only about ~3,900 SLOC for PPE/SPE-specific instructions
    - ~2,300 SLOC to use them in PARTISN-proper
  - Major changes:
    - The structure of the algorithms was retained, but they were rewritten to allow for “threading” and vectorization
    - Data structures were modified so that they would fit on the SPE and to enable vectorization
  - Roadrunner specific coding has been removed, but…
    - The new vectorized ‘cell branch’ is 2+ times more performant on today’s commodity machines
    - Formed the basis for PARTISN’s MPI+OpenMP implementation
      - MPI decompose in space, thread in energy, vectorize in angle

Roadrunner – 2009

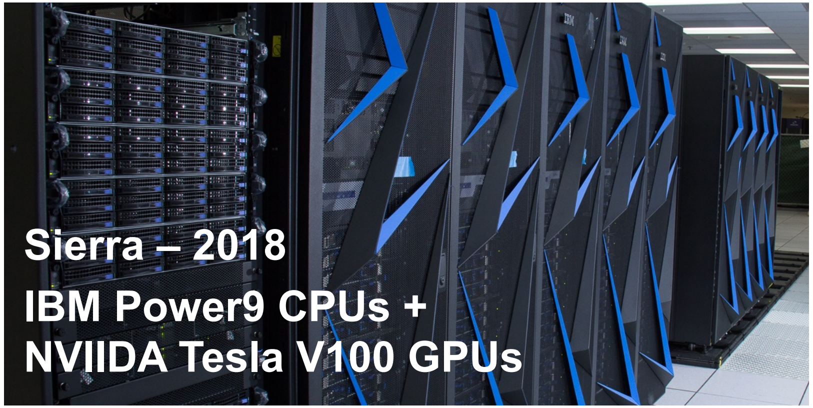
AMD Opteron/IBM cell accelerator (PPE/SPE)



Modified from slides by Randy Baker

“Mature” Fortran Code Meets Advanced Architectures

- The initial port to Sierra was completed in 2020
- Production-ready release planned for summer 2021
  - Randy Baker, Joe Zerr, Dan Magee (HPC-ENV), Nate Hart, Kris Garrett
  - 5 developers over 2.5 years for a total of 3-4 FTEs
- Starting point was the Roadrunner algorithm
  - ~4400 SLOC, CUDA Fortran
- And then...Crossroads (2022) and El Capitan (2023)



- PARTISN’s porting efforts for Roadrunner and Sierra have involved rewriting and hand-tuning compute-intensive pieces of the code and data movement between hardware components
- “Transport Sweeps” are the core algorithm for S_N transport—Understanding the hardware and tailoring it’s implementation for that hardware has been a highly successful strategy for achieving performance on advance architectures

Thoughts on what has made PARTISN successful

- Willingness to learn and evolve
  - Code team are nuclear engineers with expertise in transport methods—Not computer scientists or software engineers by training!
  - In-depth knowledge of (1) how the algorithms work and (2) the application space
  - Willing to dig deep to learn how our software can best be adapted to ever-evolving HPC resources—*Take the time to develop knowledge of the hardware as well!*
- Engage with CS experts
  - Mini-app (SNAP) for engaging with vendors
  - Built collaborations with LANL HPC staff, Cray, NVIDIA, Intel
- Engage with procurements and Centers of Excellence
  - This is a lot of work, but keeps us up to date and allows us to (to some extent) drive decisions about the machines that PARTISN has to run on
- Always thinking ahead to the next architecture
  - Take the lessons from the past and use them to inform future designs

Looking forward

- Six team members (5 staff, 1 postdoc) who spend varying amounts of time working on the code and supporting LANL users (~3.5 FTEs)
  - Methods development—Just because the code is old, that does not mean it is complete!
    - Using machine learning to improve multigroup data *Tom Saller*
    - Self-shielding corrections for multigroup data *Tom Saller*
    - Solutions to the stochastic neutronics equations *Erin Davis*
    - Use of Block Adaptive Mesh Refinement (AMR) *Mario Ortega*
    - Efficient and scalable computational transport solution methods for massively parallel architectures using a “MPI+X” paradigm *Joe Zerr, Nate Hart*
  - Software maintenance, modernization and bug fixes
  - User support—Help our users run the code *and* understand their results
- What does our future hold?
  - Will Fortran remain a viable programming language for HPC physics applications???
  - Will retaining control over the parallelization and data movement remain a viable strategy for performance portability???
  - There is a lot of code modernization that should to be done if PARTISN will be actively developed for 10+ years
  - *In the end, LANL will always have a deterministic transport code—The knowledge and experience embodied by the team is of most value to LANL and our user community*