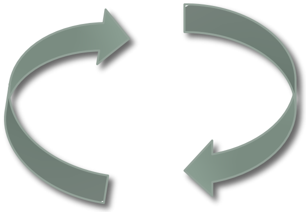
OVERVIEW OF CESAR PROXY APPS

Andrew Siegel, Deputy Director
Kord Smith, Chief Scientist
Center for Exascale Simulation of Advanced Reactors (CESAR)

CESAR goals

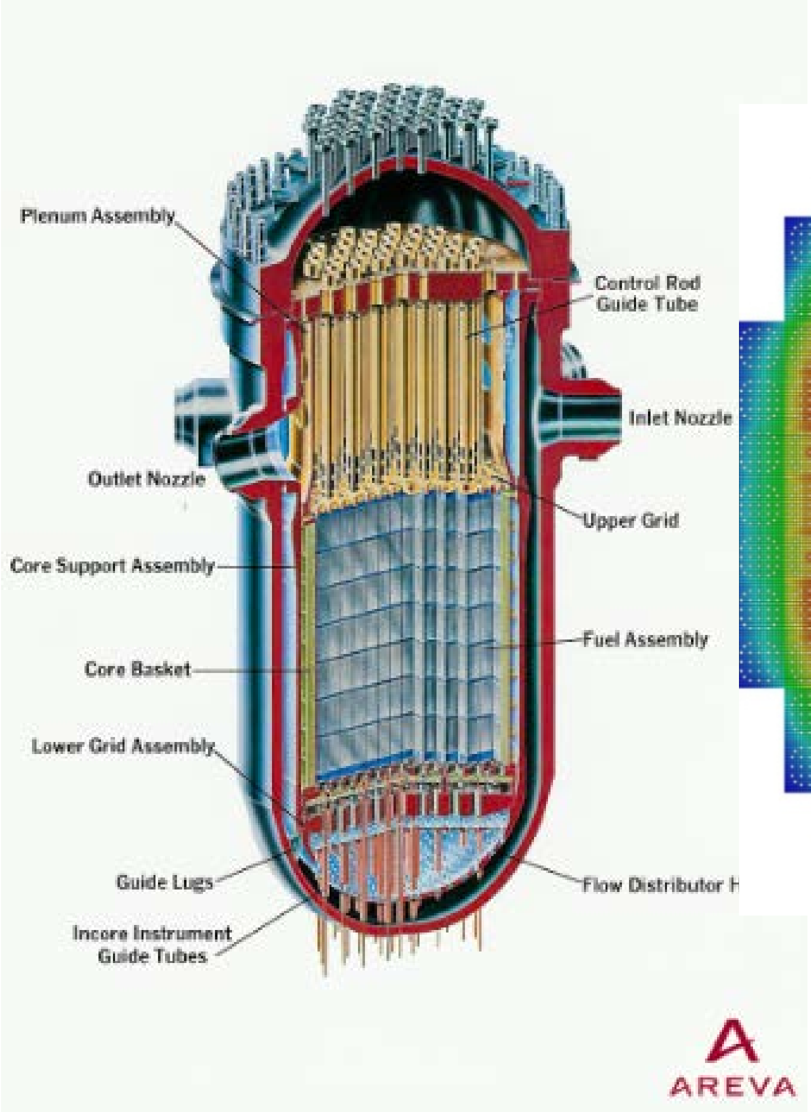
Developing algorithms to enable efficient reactor physics calculations on exascale computing platforms.

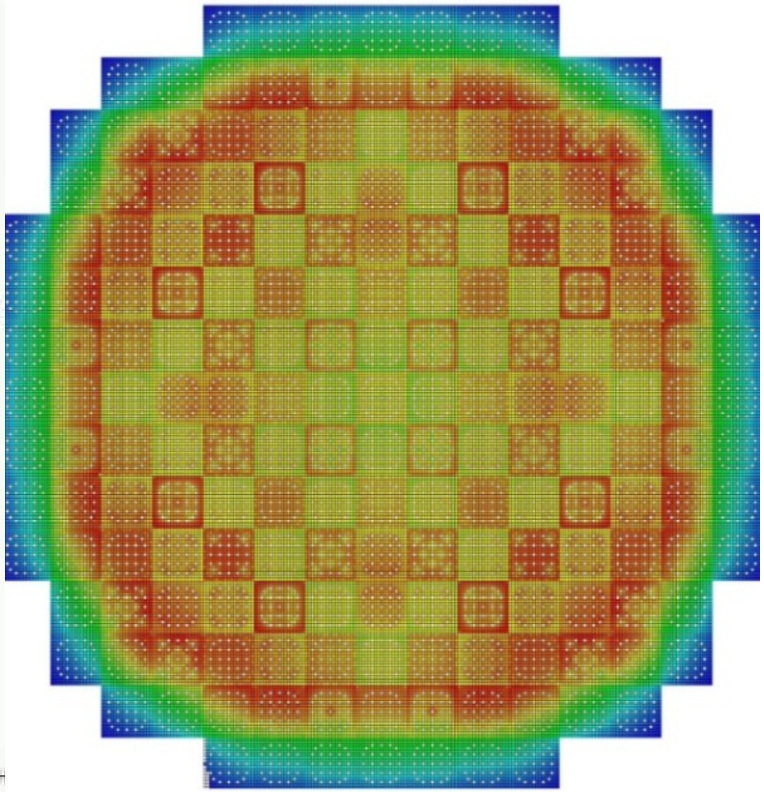


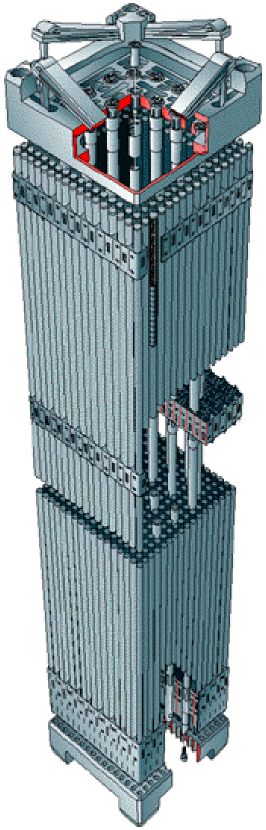
Influencing exascale hardware/x-stack priorities, innovation based on “needs” key algorithms

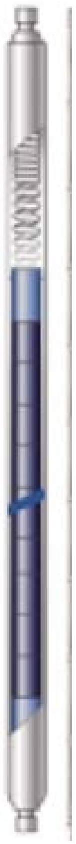
- It is assumed in this discussion that exascale capability is needed for nuclear energy industry and that next-generation reactors are strategically important.

CESAR Challenge: Predict Pellet-by-Pellet Power Densities and Nuclide Inventories for the Full Life of Reactor Fuel (~5 years)









CESAR Applications

Nek

Computational Fluid Dynamics → Incompressible Navier-Stokes → Spectral Elements

- High FLOP/load ratios
- Nearest neighbor
- Bulk synchronous
- Low memory per node required for scalability
- Global AllReduce latency key

UNIC

Neutron Transport → Boltzmann → Method of Characteristics

- tabulated exponential
- Complex parallelization
- Load balancing issues
- High FLOP/s rate

OpenMC

Neutron Transport → Stochastic (Monte Carlo) → Data and Domain Decomposition

- Load dominated
- Branch heavy
- Highly parallelizable in particle space
- Poor locality in x-section and tally space
- Low FLOP/s rate
- Performance hot spot

All three → Mini-Apps, Kernels

Proxy Apps

Mini-apps: reduced versions of applications intended to …

- Enable communication of application characteristics to non-experts
- Simplify deployment of applications on range of computing systems
- Facilitate testing with new programming models, hardware, etc.
- Serve as a basis for performance model, profiling

- Must distinguish between code and application of code
  - One key for mini-app is to appropriately constrain problem, input etc.
  - We all worry about abstracting away important features

- For CESAR the three key mini-apps are
  - *Nek-bone*: spectral element poisson equation on a square
  - *MOC-FE*: 3d ray tracing (method of characteristics) on a cube
  - *mini-OpenMC*: Monte Carlo transport on a pre-built simplified lattice
  - TRIDENT: transport/cfd coupling, still under development

- Algorithmic innovations for exascale embedded in **kernel apps**:
  - MCCK, EBMS, TRSM, etc.

https://cesar.mcs.anl.gov/content/software

What Is the Scale of Monte Carlo LWR Problem?

~200	fuel **assemblies**
~70,000	discrete fuel **pins**
~35,000,000	discrete fuel **pellets**
~350,000,000	discrete depletion **zones**
~1,000,000,000,000	bytes of tally data for 300 **nuclides**
~100,000,000,000,000	bytes of tally data for complete **fuel history**

- State of the art MC codes can perform single-step depletion with 1% statistical accuracy for 7,000,000 pin power zones in ~100,000 core-hours.

- What is needed for Exascale Application of Monte Carlo LWR Analysis?
  - Efficient on-node parallelism for particle tracking (70% scalability on up to 48 cores per node but wide variation and possible limitations)
  - The ability to execute efficiently with non-local **1 T-byte data tallies**
  - The ability to access very **large x-section lookup tables** efficiently during tracking
  - The ability to treat **temperature-dependent** cross sections data in each zone
  - The ability to **couple to detailed fuels/fluids** computational modeling fields
  - The ability to **efficiently converge** neutronics in non-linear coupled fields
  - Capability of **bit-wise reproducibility** for licensing: data resiliency model key

Co-design opportunities for Temperature-Dependent Cross Sections

- **Cross section data size:**
  - **~2 G-byte for 300 isotopes at one temperature**
  - **~200 G-byte for tabulation over 300K-2500K in 25K intervals**
    - **Data is static during all calculations**
    - **Exceeds node memory of anticipated machines**

- **Represent data with discrete temperature approximate expansions?**
  - **New evidence that 20-term expansion may be acceptable**
  - **~40 G-byte for 300 isotopes**
    - **Large manpower effort to preprocess data**
    - **Many cache misses because data is randomly accessed during simulations**

- **NV-Ram Potential?**
  - **Data is static during all simulations**
    - **Size NV-RAM needed depends on data tabulation or expansion approach**
    - **Static data beckons for non-volatile storage to reduce power requirements**
    - **Access rate needs to be very high for efficient particle tracking**

Co-design Opportunities for Large Tallies

- Spatial domain decomposition?
  - Straightforward to solve tally problems with limited-memory nodes
  - Communication is 6-node nearest-neighbor coupling
    - Small zones have large neutron leakage rates –> implications for exascale
    - Using a small number of spatial domains may allow data to fit in on-node memory
    - Communications requirements may be significant
- Tally-server approach for single-domain geometrical representation?
  - Relatively small number of nodes can be used as tally servers
  - Each tally server stores a small fraction of total tally data
  - Asynchronous writes eliminate tally storage on compute nodes
  - Compute nodes do not wait for tally communication to be completed
    - Local node buffering may be needed to reduce communication overhead
    - Communications requirements may be still be significant
    - Global communication load may become the limiting concern

Co-design opportunities for Temperature-Dependent Cross Sections

- **Direct re-computation of Doppler broadening?**
  - **Cullen's method to compute cross section integral directly from 0^0K data, or**
  - **Stochastically sample thermal motion physics to compute broadened data**
    - **Never store temperature-dependent data, only the 0^0K data**
    - **Cache misses will be much smaller than with tabularized data**
    - **Flop requirement may be large, but it is easily vectorizable**

- **Energy domain decomposition?**
  - **Split energy range into a small number (~5-20) energy "supergroups"**
  - **Bank group-to-group scattering sites when neutrons leave a domain**
  - **Exhaust particle bank for one domain before moving to next domain**
  - **Use server nodes to move cross section only for the active domain**
    - **Modest effort to restructure simulation codes**
    - **Cache misses will be much smaller than with full range tabularized data**
    - **Communication requirements can be reduced by employing large particle batches**