# Preliminary Assessment of Parallel Efficiency of SCALE CSAS6 and T6-DEPL Sequences

**Bojan Petrovic, Kyle Carberry, Jonathon Faulkner, Christopher Bayne**
Georgia Institute of Technology, Nuclear Engineering Program
770 Ferst St., Atlanta, GA 30332-0745, USA
bojan.petrovic@gatech.edu, kcarberry3@gatech.edu, jfaulkner31@gatech.edu
christopher.bayne@gatech.edu

**Ronald Rahaman**
Georgia Institute of Technology, Partnership for an Advanced Computing Environment (PACE)
756 West Peachtree Street NW, Atlanta, GA 30332-0700, USA
rrahaman6@gatech.edu

## ABSTRACT

Reactor physics analyses of complex nuclear systems simulations aiming to achieve accuracy rely to a large degree on Monte Carlo methods. The challenge then becomes reducing the statistical uncertainty which generally requires simulating a large number of neutron histories. Achieving required statistical convergence within acceptable turnaround time requires in most cases parallel simulations.

The SCALE code suite developed and maintained by Oak Ridge National Laboratory (ORNL) is presently composed of 11 end user products with capabilities ranging from radiation shielding to sensitivity and similarity analysis. Of particular interest for reactor design is KENO-VI continuous energy (and multigroup) Monte Carlo radiation transport code with corresponding sequences for criticality and depletion evaluation, CSAS6 and T6-DEPL. SCALE 6.2 contains several modules and sequences that have distributed memory (MPI) parallelism, including KENO-VI. SCALE 6.2.4 with these parallel capabilities has been successfully built and installed on the Georgia Tech PACE cluster.

The paper reports results of preliminary testing of parallel performance for several representative problems, from simple to more complex ones, from static no depletion to depletion cases, evaluated for weak and strong scaling, on a single multi-CPU node as well as on multiple nodes. The simple Godiva problem achieves a maximum strong scaling speedup of about 7, and this does not improve significantly for the weak scaling, suggesting that the inherent bottleneck is in very limited computational effort required per particle history. SCALE parallel diagnostics provides useful data and supports this conclusion. The uranyl nitrate solution test problem provides better parallel performance due to longer neutron histories, but is still too simple to significantly benefit. More complex MSR depletion problem (T6-DEPL sequence) achieves around 80% parallel efficiency on 24-96 CPUs. Finally, a more relevant problem representing a complex FHR fuel assembly geometry achieves parallel efficiency exceeding 90% on single node and multiple nodes, tested on up to 96 CPUs. Used cautiously, these findings can provide a useful a priori indication of possible speedup and a guidance how to improve it.

## 1 INTRODUCTION

Reactor physics analyses of complex nuclear systems simulations aiming to achieve best possible accuracy rely to a large degree on Monte Carlo methods. Traditionally performed only for

benchmarking studies, they are now increasingly used for reference calculations, and in some cases even for everyday analyses. The computational challenge is to reduce the statistical uncertainty which generally requires simulating a large number of neutron histories. Achieving required statistical convergence within acceptable turnaround time requires in most cases parallel simulations.

The SCALE code suite developed and maintained by Oak Ridge National Laboratory (ORNL) is presently composed of 11 end user products with capabilities ranging from radiation shielding to sensitivity and similarity analysis[1]. Of particular interest for reactor design is KENO-VI continuous energy (and multigroup) Monte Carlo radiation transport code with corresponding sequences for criticality and depletion evaluation, CSAS6 and T6-DEPL. SCALE 6.2 contains several modules and sequences that have distributed memory (MPI) parallelism, including KENO-VI. This paper presents preliminary assessment of parallel efficiency of SCALE CSAS6 and T6-DEPL sequences.

## 2 CODES AND COMPUTATIONAL RESOURCES

### 2.1 SCALE 6.2.4 Parallel Capability

SCALE 6.2 contains four modules/sequences with distributed memory (MPI) parallelism. However, the binary executable files distributed with SCALE do not have MPI enabled and the user needs to build an MPI-enabled executable. This process is not always trivial and usually requires adjustment to account for the local configuration specifics. Additionally, parallel version is not available for Windows.

### 2.2 Installing MPI-enabled SCALE 6.2.4 on Georgia Tech PACE

Georgia Tech high performance computing (HPC) capability is provided through Partnership for an Advanced Computing Environment (PACE). Computational resources include several clusters with a total of over 2,000 nodes and 50,000 CPU total. Install was performed on a smaller cluster (~50 nodes, ~1,000 CPU, 192/284/762 GB RAM per node).

To compile SCALE 6.2.4, we first required a separate compiler and library stack to match the significantly older versions used by the SCALE developers for testing and release (GCC 4.8.5 and OpenMPI 1.8.5). While this older compiler/library stack had been deprecated at PACE, we now believe that maintaining this stack is a beneficial long-term effort, since it has already simplified our builds of the SCALE 6.3 beta release. We successfully used BLAS/LAPACK implementations from OpenBLAS but were not successful in our attempts to use Intel MKL, which is highly optimized for Intel CPUs and was a prospective optimization technique for SCALE. Compiling with OpenMP support was straightforward and was completed with no complications.

### 2.3 Evaluation of Parallel Scaling

Testing was performed using Moab scheduler along with the Torque resource manager, on up to 100 CPU aimed to represent the computational resources of most users. Specifically, parallel simulations were performed on a single CPU to assess the serial time, then on 1, 2, 3 and 4 nodes with 24 CPU each, i.e., on 24, 48, 72 and 96 CPU respectively. Both strong and weak parallel scaling was evaluated. In the former, the problem size is kept constant; in the latter, the problem size is increased proportionally with the number of CPU.

## 3 MODELS

It is expected that parallel performance will depend on many simulation parameters and characteristics, including the reactor physics characteristics of the problem being modelled. Four representative problems were selected to cover a broad range of practical applications:

1. Godiva critical experiment (highly enriched metal uranium bare sphere) – essentially the simplest possible criticality simulation.
2. Highly enriched uranyl nitrate solution – representing a simple criticality safety problem, but with more complex reactor physics due to the presence of water moderator.
3. Fuel element of fluoride-salt-cooled high-temperature reactor (FHR) – representing a reactor design simulation with a very complex geometry.
4. Molten salt reactor (MSR) depletion – representing fuel cycle simulation.

The first three cases employed CSAS6 sequence (criticality analysis sequence using KENO-VI), while depletion used T6-DEPL (CSAS6 combined with ORIGEN). The first two cases are fairly well known and frequently used for verification and validation (V&V); a detailed description may be found in [2]. FHR fuel element used in this study is described in [3] and [4] and shown in Figure 1. The MSR case is intended to represent a typical thermal MSR moderated by graphite and fueled by FLiBe with $UF_4$. A simplified model of the Molten Salt Research Reactor (MSRR) [5], as shown in Figure 2, was employed.

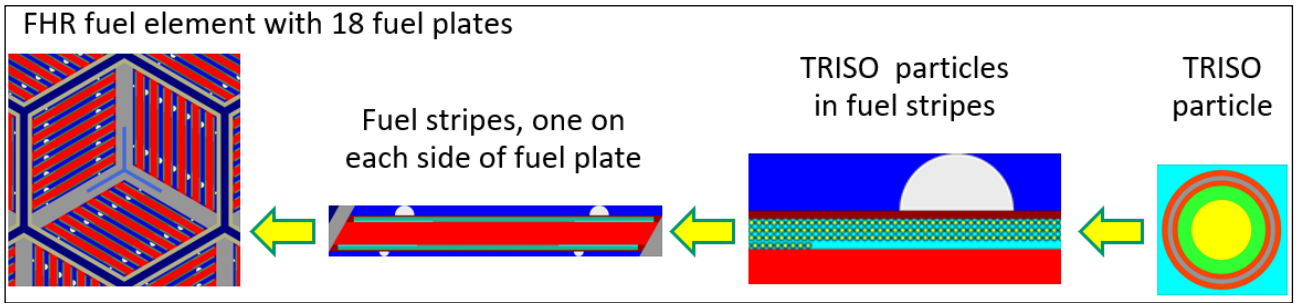


Figure 1: FHR fuel element

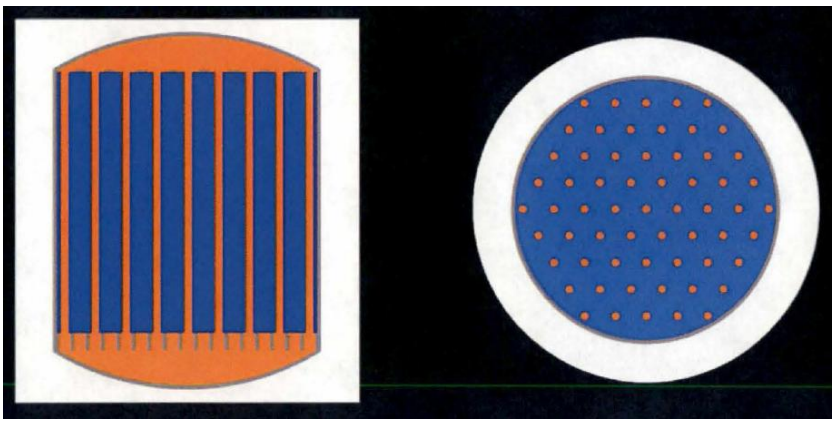

Figure 2: MSRR simplified model

## 4 RESULTS AND ANALYSIS

All simulations were performed using the continuous energy (CE) ENDF/B-VII.1 library [6]. For each case the number of cross section sets (i.e., number of distinct materials) that needed to be prepared is indicated together with the number of skipped and total generations, and the number of histories per generation. Number of nodes is given together with the number of CPU, which is the number of nodes multiplied by 24 (CPU per node). A serial run (with a single CPU) was performed in each case and used to calculate the speedup and parallel efficiency, defined in usual way:

(1)

(2)

### 4.1 Godiva Critical Experiment

Results of parallel performance testing are shown in Table 1. Both strong and weak scaling was examined. For strong scaling test cases, number of histories per generation was kept at 100,000. For weak scaling cases, the problem size was increased only going from 24 CPU to higher, but not from 1 to 24 CPU.

Table 1: Parallel efficiency for Godiva simulations

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Library | CE | | | | | | | |
| Number of Cross Sections to Process | 3 | | | | | | | |
| Number of Generations | 400 | | | | | | | |
| Number of Generations Skipped | 40 | | | | | | | |
| | | | | | | | | |
| | Case 10 | Case 11 | Case 12 | Case 13 | Case 14* | Case 15 | Case 16 | Case 17 |
| Number of Nodes | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Number of Processors (Total) | 24 | 48 | 72 | 96 | 24 | 48 | 72 | 96 |
| Number of Histories per Generation | 100000 | 100000 | 100000 | 100000 | 100000 | 200000 | 300000 | 400000 |
| Parallel Performance | | | | | | | | |
| Speedup | 6.96 | 7.55 | 7.75 | 7.34 | 6.96 | 7.81 | 7.49 | 8.15 |
| Parallel Efficiency | 30.25 | 16.06 | 10.91 | 7.73 | 30.25 | 16.61 | 10.55 | 8.58 |
| Wall Time (min) | | | | | | | | |
| Serial Code Section | 0.7031 | 0.7492 | 0.7685 | 0.7797 | 0.7031 | 1.4536 | 2.4185 | 2.9783 |
| Parallel Code Section | 0.8203 | 0.3999 | 0.2674 | 0.2006 | 0.8203 | 0.8158 | 0.7929 | 0.805 |
| Communication Interface | 1.3092 | 1.4506 | 1.5372 | 1.7393 | 1.3092 | 2.8467 | 4.6475 | 6.0005 |
| Total | 2.8128 | 2.5892 | 2.5502 | 2.7017 | 2.8128 | 5.0963 | 7.8393 | 9.7464 |

* Case 14 is identical to Case 10

Relatively poor parallel efficiency is observed in all cases. For strong scaling (Cases 10 through 13), adding more CPU does not help; the total wall clock time remains the same, and parallel efficiency drops from 30.25% for 24 CPU to 7.55% for 96 CPU. This behaviour should not

come as a surprise and does not at all reflect SCALE parallel capability. It is driven by the reactor physics of the problem. Godiva is a small, fast critical system, with high leakage. Fission neutrons are likely to quickly (with small number of interactions) leave the system or cause another fission. Computational effort per history is minimal. The total run time for serial run (not shown in the table) is only about 20 min. There is no practical need to split a 20-min run over 24 CPU. Weak scaling (Cases 14 through 17) only marginally helps. However, if one needed to generate a solution with extremely low statistical uncertainty (e.g., for code cross-verification), with a significant associated increase in the number of histories per generation (e.g., by 2 or more orders of magnitude), parallel performance would improve.

### 4.2 Highly enriched uranyl nitrate solution

Results of parallel performance testing are shown in Table 2. Both strong and weak scaling was examined. For strong scaling (Cases 50 through 53), number of histories per generation was kept at 100,000. For weak scaling (Cases 54 through 57), the problem size was increased only going from 24 CPU to higher, but not from 1 to 24 CPU.

This problem presents a more complex reactor physics. Water moderator and reduced leakage result in significantly longer neutron histories. With the same number of generations and histories per generation, serial run takes about 16 times longer and benefits more from parallel execution, resulting in a reasonable efficiency of 82.85% on 24 CPU. Adding further CPU becomes more difficult to justify; quadrupling the number of CPU, from 24 to 96, increases the speedup by only about 50%.

Table 2: Parallel efficiency for highly enriched uranyl nitrate solution simulations

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Library | CE | | | | | | | |
| Number of Cross Sections to Process | 39 | | | | | | | |
| Number of Generations | 400 | | | | | | | |
| Number of Generations Skipped | 40 | | | | | | | |
| | | | | | | | | |
| | Case 50 | Case 51 | Case 52 | Case 53 | Case 54* | Case 55 | Case 56 | Case 57 |
| Number of Nodes | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Number of Processors (Total) | 24 | 48 | 72 | 96 | 24 | 48 | 72 | 96 |
| Number of Histories per Generation | 100000 | 100000 | 100000 | 100000 | 100000 | 200000 | 300000 | 400000 |
| Parallel Performance | | | | | | | | |
| Speedup | 19.06 | 22.83 | 26.98 | 29.51 | 19.06 | 22.69 | 27.44 | 30.54 |
| Parallel Efficiency | 82.85 | 48.58 | 38.00 | 31.06 | 82.85 | 48.27 | 38.65 | 32.14 |
| Wall Time (min) | | | | | | | | |
| Serial Code Section | 1.6405 | 6.0314 | 6.1211 | 6.1804 | 1.6405 | 12.4094 | 18.0585 | 23.5967 |
| Parallel Code Section | 14.1003 | 6.8982 | 4.5852 | 3.4483 | 14.1003 | 13.8057 | 13.7491 | 13.6792 |
| Communication Interface | 1.384 | 1.5448 | 1.6095 | 1.7035 | 1.384 | 2.9545 | 4.4602 | 6.073 |
| Total | 17.1046 | 14.4646 | 12.2922 | 11.3121 | 17.1046 | 29.1498 | 36.2309 | 43.3303 |

* Case 54 is identical to Case 50

### 4.3 FHR Fuel Element

Results of parallel performance testing are shown in Table 3. The problem has fairly complex geometry and simulation in the serial mode takes about 10 days. Due to such long execution time, it was not practical to cover the same testing matrix as for the previous two problems. Case 71 on 24 CPU achieves excellent parallel efficiency exceeding 97%. Doubling the number of CPUs while keeping the problem size the same (Case 72) produces slightly lower efficiency, as expected, but still exceeding 94%. Case 73 uses more nodes/CPU, while reducing the number of histories in half. Thus, parallel efficiency is double penalized. Nevertheless, it achieves commendable 91%. Finally, Case 74 further increases the number of nodes from 3 to 4 (CPU from 72 to 96). Parallel efficiency is 88%, i.e., there is some additional, but still quite acceptable, decrease of efficiency. The 10-day serial run on 96 CPU takes only about 3 hours, making such type of analysis significantly more viable.

Table 3: Parallel efficiency for FHR fuel element simulations

| | | | | |
|---|---|---|---|---|
| Library | CE | | | |
| Number of Cross Sections to Process | 14 | | | |
| Number of Generations | 700 | | | |
| Number of Generations Skipped | 50 | | | |
| | Case 71 | Case 72 | Case 73 | Case 74 |
| Number of Nodes | 1 | 2 | 3 | 4 |
| Number of Processors (Total) | 24 | 48 | 72 | 96 |
| Number of Histories per Generation | 100000 | 100000 | 50000 | 50000 |
| Statistical uncertainty pcm | 10 | 10 | 14 | 18 |
| Parallel Performance | | | | |
| Speedup | 22.41 | 44.43 | 64.64 | 83.62 |
| Parallel Efficiency | 97.46 | 94.54 | 91.05 | 88.02 |
| Wall Time (min) | | | | |
| Serial Code Section | 14.0513 | 14.494 | 8.3987 | 8.2363 |
| Parallel Code Section | 627.7362 | 307.5604 | 108.315 | 79.55 |
| Communication Interface | 3.0081 | 3.6346 | 2.3973 | 2.8375 |
| Total | 644.75 | 325.6632 | 119.0984 | 90.4721 |

### 4.4 MSR Depletion

Results of parallel performance testing are shown in Table 4. This is computationally even more intense problem than the previous one. At the same time, depletion calculations alternating with transport simulations are expected to negatively impact parallel efficiency. As before, due to long run times, only several cases have been executed. Case 81 and 82 used very small number of histories (only 1,000 histories per generation, and only 100 generations) so the low parallel efficiencies (11% and 39%) are expected. It is interesting however that Case 81 employing CE cross sections exhibits significantly lower parallel efficiency that Case 82 employing SCALE standard 252-group library. The next 3 cases (83, 84 and 85) use more realistic 40,000 histories per each of the 400 generations, and employ in turn CE, 252-group and 56-group cross section libraries. While the multigroup simulations are significantly faster (13.7 and 18.6 times), parallel

performance is in all cases fairly good (parallel efficiency above 75%) and similar (parallel efficiencies 75.74%, 79.08% and 78.13%).

Table 4: Parallel efficiency for MSR depletion simulations

| | | | | | |
|---|---|---|---|---|---|
| Library | CE | 252 | CE | 252 | 56 |
| Number of Cross Sections to Process | 40 | 40 | 40 | 40 | 40 |
| Addnux | 4 | 4 | 4 | 4 | 4 |
| Number of Generations | 100 | 100 | 400 | 400 | 400 |
| Number of Generations Skipped | 3 | 3 | 35 | 35 | 35 |
| Number of Steps | 15 | 15 | 15 | 15 | 15 |
| | | | | | |
| | Case 81 | Case 82 | Case 83 | Case 84 | Case 85 |
| Number of Nodes | 1 | 1 | 3 | 1 | 1 |
| Number of Processors (Total) | 24 | 24 | 72 | 24 | 24 |
| Number of Histories per Generation | 1000 | 1000 | 40000 | 40000 | 40000 |
| Parallel Performance | | | | | |
| Speedup | 2.54 | 8.95 | 53.78 | 18.19 | 17.97 |
| Parallel Efficiency | 11.04 | 38.93 | 75.74 | 79.08 | 78.13 |
| Wall Time (min) | | | | | |
| Serial Code Section | 13.9037 | 0.0315 | 17.3258 | 0.3622 | 0.2888 |
| Parallel Code Section | 1.1990 | 0.0208 | 60.1456 | 3.2532 | 2.5693 |
| Communication Interface | 1.2380 | 0.0047 | 2.2887 | 0.5254 | 0.4519 |
| Total Simulation Time (per step) | 16.3389 | 0.0568 | 79.7323 | 4.1337 | 3.3044 |
| Total | 262.9250 | 29.0000 | 1277.9000 | 93.4800 | 68.7250 |

## 5 CONCLUSIONS

The paper reports results of preliminary testing and examines parallel performance of SCALE 6.2.4 for several representative problems, from simple to more complex ones, from static no depletion to depletion cases, evaluated for weak and strong scaling, on a single node as well as on multiple nodes. The simple Godiva problem achieves a speedup of only about 7 with 96 CPU, and this does not improve significantly for the weak scaling, suggesting that the inherent bottleneck is in very limited computational effort required per particle history. SCALE parallel diagnostics provides useful data and supports this conclusion. The uranyl nitrate solution test problem provides better parallel performance due to longer neutron histories, but is still too simple to significantly benefit. A more relevant problem representing a complex FHR fuel assembly geometry achieves parallel efficiency exceeding 90% on single node and multiple nodes, tested on up to 96 CPUs. More complex MSR depletion problem (T6-DEPL sequence) achieves around 80% parallel efficiency on 24-72 CPUs. Used cautiously, these findings can provide a useful a priori indication of possible speedup and a guidance how to improve it. The overarching finding is that SCALE 6.2.4 performs fairly well for computationally challenging problems, while—as expected—it may experience degradation of parallel efficiency for less complex problems, in particular with very short particle histories. For practical purposes, such behaviour in most cases provides the required parallel performance, as the former (i.e., challenging) types of problem are the ones where efficient parallel simulations are needed.

## 6 ACKNOWLEDGMENTS

This research was supported, in part, by Natura Resources, LLC, and through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA

## 7 REFERENCES

[1] W. A. WIESELQUIST, R. A. LEFEBVRE, and M. A. JESSEE, Eds., SCALE Code System, ORNL/TM-2005/39, Version 6.2.4, Oak Ridge National Laboratory, Oak Ridge, TN, 2020.

[2] "IRPhe Handbook 2020", *International Reactor Physics Evaluation Project Handbook* (database), Nuclear Energy Agency, Paris, 2020 https://doi.org/10.1787/d863e360-en (accessed on 29 August 2022).

[3] K. RAMEY, B. PETROVIC, "Monte Carlo modeling and simulations of AHTR fuel assembly to support V&V of FHR core physics methods," *Annals of Nuclear Energy*, Vol. 118, pp. 272-282, 2018.

[4] B. PETROVIC, K. RAMEY, I. HILL, "Benchmark Specifications for the Fluoride-salt High-temperature Reactor (FHR) Reactor Physics Calculations," NEA-2021, Nuclear Energy Agency, 2021.

[5] Abilene Christian University – Submittal of the NEXT Lab Molten Salt Research Reactor Licensing Regulatory Engagement Plan (08/2020). [Document ML20241A071 retrieved from https://adams.nrc.gov on 01/2021]

[6] M. B. CHADWICK et al., "ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data," *Nuclear Data Sheets*, Vol. 112, pp. 2887-2996, 2011.