Testing, Simulating, and Profiling Designated Problems for CASL

Yael M. Camacho Bonaparte
Computer Engineering Department
Polytechnic University of Puerto Rico
San Juan, PR 00919

John Turner
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Abstract

Virtual Environment for Reactor Applications (VERA) is a virtual reactor for predictive simulations of light water reactors. It is currently undergoing development by the Consortium for Advanced Simulation of Light Water Reactors (CASL)[1]. Simulation is an essential step for future implementations of more efficient and safer reactor components. In nuclear technology, this translates to cost, fuel, and waste reduction, and enhanced safety.

The research project will involve activities such as testing, simulation, running designated problems on different programs or environments, analyzing, visualizing, and verifying the results, and providing feedback to the development team. The research results will provide the development team valuable information about current and future components design, predictive capabilities for possible failures, better understanding of fuel consumption, and extending reactors lifetime.

Key Words: Simulation, Testing, Profiling

.I. Introduction

Since computers were available, scientists have been modeling real-life scenarios to predict systems behavior of different parameters. Modeling and simulations are a cheaper, safer, and faster way to understand how systems behave before spending time and money on new designs or systems. Dangerous system conditions can be simulated in a totally safe environment in order to determine the real-life results.

The Consortium for Advanced Simulation of Light Water Reactors (CASL) is a partnership between the Department Of Energy (DOE) laboratories, highly qualified educational institutions, and outstanding industry companies. By implementing advanced modeling and simulation capabilities, CASL plans to reduce costs, fuel consumption, and waste while increasing safety.

CASL is developing a modeling tool called the Virtual Environment for Reactor Applications (VERA). This tool will simulate the performance of light water reactors using current and advanced modeling and simulation capabilities.

II. Equipment and Methodology

COBRA-TF (CTF) is a thermal-hydraulic subchannel code for light water reactors transient analyses developed by the Reactor Dynamics and Fuel Management Group (RDFMG)[2] and distributed to Oak Ridge National Laboratory (ORNL) for the CASL Project. The research project included activities such as testing, simulation, running designated problems on different environments, analyzing and visualizing the results.

COBRA-TF was compiled using different optimization options to get the best performance. Gprof (-pg compilation option) was used to profile the code's performance and behavior. The output was then collected, analyzed and visualized. Kprof gave us a visual look at the call graph; this helped us understand the code's behavior.

III. Results

During this specific simulation, the two most time-consuming routines are xschem and tgas. Tgas is a remarkably small routine that takes a fraction of a second to run, but it is being called around 124 million times, making it the second most time-consuming routine during the simulation. This information gave developers an idea on where to look for improving the code's performance.

%	cumulative seconds	self seconds	calls	self s/calls	total s/calls	name	Description	Called by	Calls
45.48	765.52	765.52	1122	0.00	0.00	xschem_	linearizes the momentum, continuity and energy equations	outer	intfr gssolv fillro vdrift tgas veloc sat dvdhl xtra1 dvdpv dvdhv
11.08	952.09	186.57	124634943	0.00	0.00	tgas_	Calculates vapor temperature and specific heat capacity cp of vapor	cobrai prep3d xschem post3d prop	
6.59	1063.09	111	1122	0.00	0.00	result_channel_	Prints the sub-channel results of the calculation at specified time intervals	edit	prop sat
5.84	1161.36	98.27	45936	0.00	0.00	sstemp_	calculates the steady state temperatures for fuel rods	heatin	gauss
5.81	1259.22	97.86	1122	0.00	0.00	temp_	Calculates temperature at the new time step for fuel rods and unheated conductors	heat	gauss
4.23	1330.38	71.16	1122	0.00	0.00	result_	Prints solution results at specified intervals	edit	sat
3.52	1389.56	59.18	31416	0.00	0.00	intfr_	Determines the current flow regime for the continuity mesh cell i,j and interfacial drag and wall friction factors	xschem	prop

Figure 1. Profile output with function's description and calls

DRAFT

Figure 2. Call graph for better understanding of the code's behavior

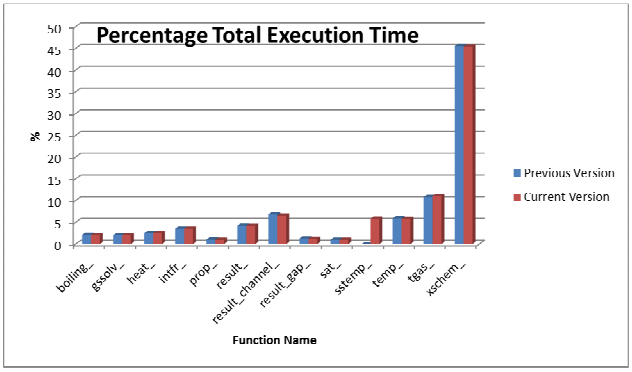

Figure 3. Difference in percentage execution time between current and previous version of COBRA-TF

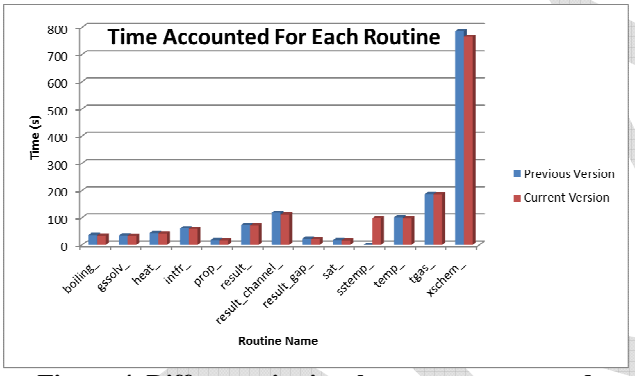

Figure 4. Difference in time between current and previous version of COBRA-TF

Figure 3 and 4 shows the differences between the current and a previous version of COBRA-TF. Both versions were profiled using the same simulation problem. Both code's versions took about the same amount of CPU time and almost the same routines were at the top of the time-consuming list. The difference in actual running time was significant; the previous version of the code took about 5.5 hours to run the simulation while the current version took only about 2.5 hours. There is an improvement of about 50% in the current version. This can be addressed as memory-use improvements on the current version.

IV. Conclusion

The research showed that the best optimization option for this simulation was –O3 with an overall performance improvement of 55.14% over no optimization (–O0). The results reported back to developers showed that 56.56% of the time is being spent at two routines; xcshem and tgas. Developers can use the results to get a better understanding of the code's behavior and performance, in order to improve the code's efficiency.

V. Future Work

Profiling can be extended to memory-use profiling with available commands or tools such as "top" or Valgrind to better understand the memory behavior during simulation. Better understanding of the memory behavior will help developers identify and fix memory leaks that can significantly affect the overall code's performance. Different input files can be simulated

VI. Acknowledgements

For the helpful advice, data, comments, and support I want to thank John Turner, Debbie McCoy, and Rashida Askia. The Research Alliance in Math and Science program is sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy. The work was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725. This work has been authored by a contractor of the U.S. Government, accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

VII. References

[1] CASL, Consortium for Advanced Light Water, http://www.casl.gov/
[2] RDFMG, Reactor Dynamics and Fuel Management Group, http://www.mne.psu.edu/rdfmg/
[3] RAMS, Research Alliance in Math and Science, http://computing.ornl.gov/internships/rams/

DRAFT