CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2012; **00**:1–22
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe

SAC/C Formulations of the All-Pairs N-Body Problem and their Performance on SMPs and GPGPUs

Artjoms Šinkarovs[1], Sven-Bodo Scholz[1*], Robert Bernecky[2], Roeland Douma[3], Clemens Grelck[3]

[1] *Heriot-Watt University, Edinburgh, UK*
[2] *Snake Island Research Inc, Toronto, Canada*
[3] *University of Amsterdam, Netherlands*

SUMMARY

This paper describes our experience in implementing the classical N-body algorithm in SAC and analysing the runtime performance achieved on three different machines: a dual-processor 8-core Dell PowerEdge 2950 (a Beowulf cluster node, the reference machine), a quad-core hyper-threaded Intel Core-i7 based system equipped with an NVidia GTX-480 graphics accelerator and an Oracle Sparc T4-4 server with a total of 256 hardware threads. We contrast our findings with those resulting from the reference C code and a few variants of it that employ OpenMP pragmas as well as explicit vectorisation.
Our experiments demonstrate that the SAC implementation successfully combines a high-level of abstraction, very close to the mathematical specification, with very competitive runtimes. In fact, SAC matches or outperforms the hand-vectorised and hand-parallelised C codes on all three systems under investigation without the need for any source code modification. Furthermore, only SAC is able to effectively harness the advanced compute power of the graphics accelerator, again by mere recompilation of the same source code. Our results illustrate the benefits that SAC provides to application programmers in terms of coding productivity, source code and performance portability among different machine architectures, as well as long-term maintainability in evolving hardware environments.

Received . . .

KEY WORDS: Single Assignment C, Data Parallelism, Functional Programming, High-Productivity, Auto-Parallelisation, Vectorisation, High-Performance, Graphics Cards

1. INTRODUCTION

The SICSA MultiCore Challenge is a long term initiative that aims at evaluating the state of the art in programming language support for multi-core systems. Since 2010, two programming challenges have been identified; researchers have been invited to contrast programming languages of their choice against a given reference implementation on a given reference system. For details on the SICSA MultiCore Challenge see [22].

This paper focuses on the second SICSA challenge, themed around the *N-body problem*. The N-body problem is that of predicting the motion of a group of celestial objects, interacting with each other gravitationally. As formulated in "Philosophiae Naturalis Principia Mathematica" by Sir Isaac

*Correspondence to: S.Scholz@hw.ac.uk, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK.