Journal of Computational Physics 310 (2016) 285–300

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Separable projection integrals for higher-order correlators of the cosmic microwave sky: Acceleration by factors exceeding 100

J.P. Briggs [a,*], S.J. Pennycook [b], J.R. Fergusson [a], J. Jäykkä [a], E.P.S. Shellard [a]

[a] Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK
[b] Intel Corporation, UK

A R T I C L E   I N F O

Article history:
Received 17 September 2015
Accepted 14 January 2016
Available online 19 January 2016

Keywords:
Cosmology
Xeon Phi
Many-core
Nested parallelism

A B S T R A C T

We present a case study describing efforts to optimise and modernise "Modal", the simulation and analysis pipeline used by the Planck satellite experiment for constraining general non-Gaussian models of the early universe via the bispectrum (or three-point correlator) of the cosmic microwave background radiation. We focus on one particular element of the code: the projection of bispectra from the end of inflation to the spherical shell at decoupling, which defines the CMB we observe today. This code involves a three-dimensional inner product between two functions, one of which requires an integral, on a non-rectangular domain containing a sparse grid. We show that by employing separable methods this calculation can be reduced to a one-dimensional summation plus two integrations, reducing the overall dimensionality from four to three. The introduction of separable functions also solves the issue of the non-rectangular sparse grid. This separable method can become unstable in certain scenarios and so the slower non-separable integral must be calculated instead. We present a discussion of the optimisation of both approaches. We demonstrate significant speed-ups of ≈100×, arising from a combination of algorithmic improvements and architecture-aware optimisations targeted at improving thread and vectorisation behaviour. The resulting MPI/OpenMP hybrid code is capable of executing on clusters containing processors and/or coprocessors, with strong-scaling efficiency of 98.6% on up to 16 nodes. We find that a single coprocessor outperforms two processor sockets by a factor of 1.3× and that running the same code across a combination of both microarchitectures improves performance-per-node by a factor of 3.38×. By making bispectrum calculations competitive with those for the power spectrum (or two-point correlator) we are now able to consider joint analysis for cosmological science exploitation of new data.

1. Introduction

The current best explanation for the origin of our universe is the inflationary big bang scenario, where it is believed that a period of exponential expansion created the large flat empty universe we see today. In addition, this model predicts that quantum fluctuations in the energy during this time will be stretched to galactic scales forming the seeds from which

* Corresponding author.
E-mail address: jb914@cam.ac.uk (J.P. Briggs).

http://dx.doi.org/10.1016/j.jcp.2016.01.019
0021-9991/© 2016 Published by Elsevier Inc.

all structure grew, from planets and stars through to super-clusters of galaxies. The statistics of these fluctuations give a window onto the dynamics at play during the birth of our universe. In particular, any deviation of these fluctuations from a Gaussian (*i.e.* Normal) distribution would be direct evidence of interesting new physics.

There has been enormous effort within the community to measure any possible deviations from Gaussianity with the bispectrum, the Fourier transform of the three point correlator, being the favoured statistic. The primary obstacle to naïve estimation of the bispectrum is that for the CMB it is 5 dimensional[1] and would require $\mathcal{O}(10^{22})$ floating point operations to calculate, which is challenging for the world's largest supercomputers. This can be overcome by using separable approximations for the bispectra, however the projection of the primordial bispectra forward to the time of observation remains a major obstacle to measurement. There are a multitude of approaches to this which divide into two main categories: ones that require non-Gaussian simulations to train estimators, [1–6]; and those that use specific simple primordial templates for which projection is tractable, [7–10].

The Modal method [11,12] developed at the University of Cambridge, which is the focus of this paper, and used by the Planck satellite experiment [13,14] is the only general method for constraining these non-Gaussianities from the available data. Its main strength is that by using a general mode expansion it can reduce the evolution from primordial to late times into a matrix projection, allowing us to constrain thousands of theoretical predictions simultaneously. By using an appropriate basis tuned for the theoretical models of interest, the Modal team have created a fast and efficient way to probe cosmological data for hints of new physics in the early universe.

This paper investigates the optimisation and modernisation of Modal, as part of an effort to accelerate it using Intel®Xeon Phi™ coprocessors. The existing MPI-level parallelism in the original code is not sufficient to enable efficient utilisation of this hardware, and we show that moving to a hybrid MPI/OpenMP implementation can significantly improve performance. For portability reasons, we avoid making any significant code changes that would benefit only one particular hardware platform, and thus the high-level programming languages and techniques that we use apply equally well to Intel®Xeon® processors. When compared to the original code on 2× processor sockets, our code optimisation efforts deliver speed-ups of 1765× on a single coprocessor and 833× on 2× processor sockets in the 2D case; and 108× on a coprocessor and 83.9× on 2× processor sockets in the 3D case. These speed-ups are large enough to significantly impact the rate of scientific discovery at COSMOS, and enable liberal use of the Modal calculation as part of future Monte Carlo pipelines – something that had previously been considered infeasible.

A number of previous studies have investigated the use of Intel Xeon Phi coprocessors to accelerate other scientific codes [15–19], and there are many similarities between the optimisations we discuss here and those explored in other domains. However, we note that many of these studies were performed before the standardisation of OpenMP 4.0 (and thus often rely on manual vectorisation via architecture-specific intrinsics). This is also the first paper (to the best of our knowledge) to explore the use of Intel Xeon Phi coprocessors for this specific application domain.

The rest of this paper is organised as follows: Section 2 provides an introduction to the two Modal routines which are being optimised – the full three dimensional calculation on the sparse non-rectangular domain, and the fast two dimensional version on a dense rectangular domain – and also provides a high-level introduction to Intel Xeon Phi coprocessors; Section 3 details the optimisation and modernisation of Modal; Section 4 presents a detailed performance study of the final application, demonstrating its scalability within a node and across multiple nodes; and finally, Section 5 concludes the paper, and discusses potential new uses for the accelerated version of the code.

2. Background

2.1. Direct integration

The primary concern of this paper is the efficient calculation of a three-dimensional inner product which concerns the projection of a set of primordial basis bispectra defined at the end of inflation into a set of basis bispectra on a spherical shell defined by the CMB; that is, we evolve from an early-time basis into another different basis which is more convenient at late times. We consider two late-time bispectra $A_{\ell_1\ell_2\ell_3}$ and $B_{\ell_1\ell_2\ell_3}$ depending on the spherical harmonic multipoles ℓ_i and we take the following inner product between them:

$$\langle A,\ B\rangle_l \equiv \sum_{\ell_i} \left(\frac{h_{\ell_1\ell_2\ell_3}}{v_{\ell_1}v_{\ell_2}v_{\ell_3}}\right)^2 A_{\ell_1\ell_2\ell_3}\,B_{\ell_1\ell_2\ell_3}\,, \tag{1}$$

where the required weight function is:

$$h^2_{\ell_1\ell_2\ell_3} = \frac{(2\ell_1+1)(2\ell_2+1)(2\ell_3+1)}{4\pi}\begin{pmatrix}\ell_1 & \ell_2 & \ell_3\\ 0 & 0 & 0\end{pmatrix}^2 . \tag{2}$$

[1] This is because it is the average of three vector quantities on a 2D surface (the CMB) which gives you 6 dimensions, enforcing momentum conservation then removes one of these.