# Hierarchical Array Layouts Require Domain-Specific Compiler Support

Olav Beckmann and Paul H J Kelly
Imperial College London
180 Queen's Gate
London SW7 2AZ
{ob3,phjk}@doc.ic.ac.uk

Hierarchical storage layouts for large multi-dimensional arrays are often better suited to making good use of memory hierarchies than traditional linear layouts such as row-major or column-major. A strength of these linear layouts is that, due to their widespread use, most compilers and dynamic execution engines in microprocessors have been targeted to optimising the traversal of such arrays with techniques such as strength reduction of array index calculations and software- or hardware-directed prefetching. As a consequence, hierarchical layouts are in danger of suffering suboptimal performance because production compilers do not understand their address calculation and microprocessor prefetching engines are unlikely to prefetch the right elements.

We argue that this problem is a specific instance of a general phenomenon: domain-specific abstractions requiring domain-specific optimisations in order to achieve good performance. In this paper, we illustrate this with examples, including array index strength reduction and software prefetching. We also discuss how hierarchical layouts as an abstraction could be implemented in some modern programming languages that provide "index" abstractions for arrays and how their use could be optimised via plug-in compiler optimisations.