Communicating Process Architectures 2007
Alistair A. McEwan, Steve Schneider, Wilson Ifill, and Peter Welch
IOS Press, 2007

Using occam-π Primitives with the Cell Broadband Engine

Damian J. DIMMICH

Computing Laboratory, University of Kent, U.K.
E-mail: damian@transterpreter.org,

Abstract. The Cell Broadband Engine has a unique non-heterogeneous architecture, consisting of an on-chip network of one general purpose PowerPC processor (the PPU), and eight dedicated vector processing units (the SPUs). These processors are interconnected by a high speed ring bus, enabling the use of different logical network topologies. When programming the Cell Broadband Engine using languages such as C, a developer is faced with a number of challenges. For instance, parallel execution and synchronisation between processors, as well as concurrency on individual processors, must be explicitly, and carefully, managed. It is our belief that languages explicitly supporting concurrency are able to offer much better abstractions for programming architectures such as the Cell Broadband Engine.

Support for running occam-π programs on the Cell Broadband Engine has existed in the Transterpreter for some time. This support has however not featured efficient inter-processor communication and barrier synchronisation, or automatic deadlock detection. We discuss some of the changes required to the occam-π scheduler to support these features on the Cell Broadband Engine. The underlying on-chip communication and synchronisation mechanisms are explored in the development of these new scheduling algorithms. Benchmarks of the communications performance are provided, as well as a discussion of how to use the occam-π language to distribute a program onto a Cell Broadband Engine's processors. The Transterpreter runtime, which already has support for the Cell Broadband Engine, is used as the platform for these experiments.

The Transterpreter can be found at www.transterpreter.org.