Assignment 11: Software Concurrent Threaded Pipelines

John Giacomoni

2006.11.27

Abstract

Traditionally, increases in transistors and fabrication technology have led to increased performance. However, these techniques are showing diminishing returns due to limitations arising from power consumption, design complexity, and wire delays. In response, designers have turned to chip multiprocessors (CMPs) that incorporate multiple cores on a single die. The performance, cost, and flexibility of these CMP systems make them appealing for threaded applications. Unfortunately, popular threading techniques require independent code regions, use expensive synchronization primitives, and use expensive communication mechanisms.

Concurrent Threaded Pipeline (CTP) architectures relax the data independence requirement and can increase computational throughput proportionately to the pipeline depth. Examples include Decoupled Software Pipelining, which focuses on compiler based extraction of pipelines from sequential codes, and the Frame Shared Memory architecture, which focuses specifically on network processing. CTP architectures show great promise for threading applications given a low-overhead high-speed blocking queue implementation.

This dissertation presents a portable general purpose software framework for realizing pipeline throughput benefits in applications amenable to being pipelined. The general applicability of the technique is confirmed along with the two software components necessary for implementation. First, a novel software-only low-overhead high-speed blocking queue implementation suitable for CTPs is presented. Second, a scheduling system for gracefully handling over-subscribed situations is described.