

บทคัดย่อ

ความท้าทายของการพัฒนาโปรแกรมบนคอมพิวเตอร์ที่มีหลายหน่วยประมวลผลบนชิปเดียว (ชิปมัลติโพรเซสเซอร์) อยู่ที่การหาวิธีดึงสมรรถนะสูงที่สุดจากทรัพยากรของระบบภายในกรอบเวลาอันจำกัด ทางออกหนึ่งที่ลดปัญหาเรื่องกรอบเวลาคือการพัฒนาโปรแกรมเชิงขนานอย่างรวดเร็วโดยใช้ส่วนต่อประสานโปรแกรมประยุกต์ ชื่อว่า โอเพนเอ็มพี (OpenMP) ที่ใช้คอมไพเลอร์สร้างเทรตและจัดการแบ่งงานโดยอัตโนมัติตามคำสั่งกำกับที่ระบุในโปรแกรม อย่างไรก็ตาม ข้อจำกัดสำคัญของโปรแกรมเชิงขนานแบบโอเพนเอ็มพีคือโปรแกรมปรับสมรรถนะได้ยากเนื่องจากสมรรถนะส่วนหนึ่งเสียไปในการประมวลผลที่คอมไพเลอร์แบ่งโดยปริยาย ซึ่งอาจจะไม่เหมาะสมกับสถาปัตยกรรมแคชทำให้เกิดเวลาสูญเปล่าเมื่อไม่พบข้อมูลที่แคชสูง แม้ว่าโอเพนเอ็มพีจะมีคำสั่งปรับลักษณะการแจกแจงให้ผู้พัฒนาโปรแกรมเลือกใช้ก็ตาม แต่การระบุคำสั่งเหล่านี้ส่งผลกระทบต่อการใช้แคชอย่างไนั้น ผู้พัฒนาโปรแกรมมักจะต้องใช้ซอฟต์แวร์ประเภทเครื่องมือวิเคราะห์สมรรถนะเพื่อช่วยค้นหาแนวทางปรับวิธีการแจกแจงงานเพื่อให้สมรรถนะสูงสุด ทั้งนี้ เครื่องมือวิเคราะห์สมรรถนะที่มีในปัจจุบันยังมีข้อจำกัดสองประการ คือ เครื่องมือที่วิเคราะห์รายละเอียดได้นั้น มักจะใช้ได้กับหน่วยประมวลผลเฉพาะบริษัทหรือเฉพาะสถาปัตยกรรม และเครื่องมือที่ครอบคลุมหน่วยประมวลผลได้หลายสถาปัตยกรรมมักจะมีส่วนวิเคราะห์การใช้งานแคชที่ยังไม่อาจจะท่อนถึงผลกระทบการเลือกคำสั่งแจกแจงได้โดยตรง วิทยานิพนธ์ฉบับนี้จึงเสนอการวิจัยและพัฒนาเครื่องมือวิเคราะห์สมรรถนะชื่อ พี-สเปา (P-SPA ย่อมาจาก Parallel program analysis using a Simics-based, Performance Analyzer) ที่สามารถใช้วิเคราะห์สมรรถนะของแคชที่สืบเนื่องมาจากการแจกแจงของโปรแกรมโอเพนเอ็มพี บนหลายสถาปัตยกรรมได้

เครื่องมือพี-สเปาพัฒนาเป็นโมดูลเสริมบนระบบจำลองซิมิกส์ (Simics) ที่ได้รับการยอมรับว่าเป็นระบบที่จำลองการทำงานของสถาปัตยกรรมคอมพิวเตอร์ได้ถูกต้อง เครื่องมือพี-สเปาเก็บข้อมูลการทำงานของแคชเมื่อประมวลผลรูปแบบการวิเคราะห์ในรูปแบบของกราฟและข้อมูลดิบ วิทยานิพนธ์ฉบับนี้นำเสนอขั้นตอนการทำวิจัยเพื่อพัฒนาและตรวจสอบความถูกต้องของเครื่องมือพี-สเปา รายงานผลการทดลองใช้พี-สเปาเพื่อวิเคราะห์สมรรถนะของชิปมัลติโพรเซสเซอร์ และปรับสมรรถนะโปรแกรมเชิงขนานรวมทั้งสิ้นหกการทดลอง ซึ่งผลสรุปโดยรวมได้ว่าเครื่องมือพี-สเปาสามารถใช้เพื่อศึกษาผลกระทบเชิงสมรรถนะของการออกแบบสถาปัตยกรรม และเพื่อหาแนวทางปรับสมรรถนะโปรแกรมเชิงขนานด้วยการเลือกวิธีแบ่งงาน ทำให้โปรแกรมเร็วขึ้นได้เป็นสัดส่วนถึงร้อยละ 10 เมื่อเทียบกับโปรแกรกดั้งเดิมที่ใช้คำสั่งอัตโนมัติตามที่คอมไพเลอร์กำหนด

Abstract

A challenge for developing programs on computers which have multiple processors on a single chip (chip-multiprocessors) is how to utilize the processor resources towards optimum performance within a time constraint. A solution to relax the time constraint is to rapidly develop parallel programs by using a standard application program interface called OpenMP. The highlight of OpenMP is that programmers only specify some directives in source programs telling compilers to generate the codes for thread creation and work sharing automatically. Despite the ease of development, OpenMP programs are difficult to tune. One reason is that the default loop scheduling generated by compilers may not fit well with particular cache architecture; resulting high ratio of cache misses. To alleviate this problem, OpenMP specification consists of several scheduling clauses for programmers to match with program characteristics. However to identify which scheduling clause is best fit to a program; programmers require some information from a performance analyzer. Recent development of performance analyzers imposes two issues. First, some analyzers which could give detail characteristics of programs are normally designed for particular processor architecture. Second, performance analyzers which cover many processor architectures normally do not relate the detail characteristics of cache usage to loop scheduling clauses. To overcome the issues, this thesis proposed the research and development of a new performance analyzer called P-SPA (Parallel program analysis using a Simics-based, Performance Analyzer). The proposed analyzer is aim to cover the performance characteristics of cache in related to scheduling clauses; and can be used to analyze various types of chip multiprocessor architectures.

The P-SPA performance analyzer implements as additional module on top of the widely acceptable, computer-architecture simulator, called Simics. The P-SPA module collects cache characteristics during an execution of a loop body and produces the analysis results in the form of graphs and data files. This thesis presents the research methodology to develop and verify the P-SPA analyzer. It also reports the studies of performance tuning using P-SPA. Six experiments have been reported to demonstrate that the P-SPA analyzer can be use as a tool for both the study on new architectural designs and on how to tune the performance of OpenMP programs. The experimental results show that using P-SPA as a tool to choose scheduling clauses, a particular workload can performs up to 10 percents faster in comparison to its original default scheduling directed by the compiler.