EXCLUSIVELY TRANSPARENT CHECKPOINTING UNDER UNIX ENVIRONMENTS

PISUT TRANCHINDAVONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
(COMPUTER SCIENCE)

IN

FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY

1996

ชื่อวิทยานิพนธ์	การทำ Transparent Checkpointing ที่สมบูรณ์ ภายใต้ระบบปฏิบัติการ Unix
ผู้วิจัย	พิสุทธิ์ ธารจินดาวงศ์
ปริญญา	วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

คณะกรรมการควบคุมวิทยานิพนธ์

ศุภชัย ตั้งวงศ์ศานต์ Ph.D.

ดำรัส วงศ์สว่าง Ph.D.

วันที่สำเร็จการศึกษา 19 ธันวาคม พ.ศ. 2539

บทคัดย่อ

เราสามารถเชื่อมโยง Workstation ให้ใช้งานแบบเครือข่ายได้และเพื่อให้เกิดประโยชน์สูงสุดน่าจะมีการแบ่งโปรแกรมจากเครื่องที่ใช้งานหนักไปทำงานในเครื่องที่มีการใช้งานน้อยกว่าโดยที่โปรแกรมดังกล่าวต้องสามารถคงสิทธิของเจ้าของเครื่องเดิมนั้นไว้ กล่าวคือ โปรแกรมต้องถอนตัวจากเครื่องทันทีที่เจ้าของเครื่องกลับเข้าสู่การใช้งาน และไปทำงานต่อในเครื่องที่ว่างต่อไป

เพื่อให้การทำงานต่อของโปรแกรมไม่ต้องเริ่มต้นใหม่ตั้งแต่ต้นโปรแกรม เราก็อาศัยวิธีการ Checkpointing ซึ่งเป็นวิธีการจัดเก็บขั้นตอนของการทำงานของโปรแกรมเป็นระยะๆ เป็นแฟ้มข้อมูลและสามารถนำแฟ้มข้อมูลดังกล่าวมาทำงานต่อจากจุดที่ถูกกระทำ Checkpoint

การวิจัยที่ผ่านมาต้อง Link โปรแกรมที่จะใช้งานเข้ากับ Checkpointing Library ซึ่งทำให้โปรแกรมมีขนาดใหญ่ขึ้นประมาณ 2 ถึง 4 เท่า นอกจากนั้นผู้ใช้งานต้องรู้ว่าจะใช้ Library, Compiler Options อะไรบ้าง ในบางวิจัยถึงกับต้องเปลี่ยนชื่อ main() ให้เป็นชื่อที่ระบุไว้ วิทยานิพนธ์นี้ได้เสนอแนวทางการทำ Checkpointing โดยไม่ต้องมีการ Link ซึ่งแสดงให้เห็นถึงการใช้งานง่าย (Simplicity) และการใช้เนื้อที่ Disk อย่างมีประสิทธิภาพ (Disk Space Utilization)

โปรแกรมการทดสอบได้ถูกสร้างขึ้นภายใต้ระบบปฏิบัติการ Linux โดยการอาศัย Background Process ทำหน้าที่ในการจัดการ Checkpointing ด้วยข้อดีของการ Dump ของระบบปฏิบัติการแบบ Unix ผลการทดสอบสามารถสร้าง Checkpoint File จาก Dump และสามารถทำงานต่อไปได้ นั่นก็คือ การทดสอบได้สนับสนุนการใช้งานง่าย (Simplicity) และเพิ่มประสิทธิภาพการใช้เนื้อที่ Disk (Disk Space Utilization)

Thesis Title	Exclusively Transparent Checkpointing under Unix Environments
Name	Pisut Tranchindavong
Degree	Master of Science (Computer Science)
Thesis Supervisory Committee	
	Supachai Tangwongsan, Ph.D.
	Damras Wongsawang, Ph.D.
Date of Graduation	19 December B.E. 2539 (1996)

ABSTRACT

Higher performance personal workstations can be connected in a cluster to represent a considerable computing resource. To take full advantage of these workstations, it should allow a process from one workstation to execute on somewhat idle workstation. Furthermore, foreign process must maintain the ownership of workstation. That is, it has to vacate itself when the workstation's owner continues the work and resumes the execution on another available workstation.

In order not to resume execution from the beginning of program every time, checkpointing is utilized. Checkpointing is a mechanism that periodically saves an image of process in memory and creates the checkpoint file that is ready to be executed from the point where it was checkpointed.

Previous approaches are required the application program relinks with checkpointing library which causes the program approximately two to four times bigger than the original executable program. The users need to know which library needs to be linked with their programs. In order to create a checkpoint file, some implementations need compiler options while some need to rename main() to specified name. This thesis proposes another way which has no library linking with the application program and its main consideration is *simplicity* or *transparency*. Since only the checkpoint files created will hold disk space and they will be erased after the program finishes execution, this is another consideration and refers to *disk space utilization*.

A prototype is developed to run on Linux machine. It bases on a background process that takes care of checkpointing. The result shows that under dump feature in Unix we can create a checkpoint file from the dump file and have it executed successfully. Therefore in conclusion, it conforms to both simplicity and disk space utilization as proposed.