

**UNIVERSITY OF MOSUL
COLLEGE OF COMPUTER SCIENCES
AND MATHEMATICS**



Design and Implementation of Process Migration at Binary Code Level

Nadia Tarik Saleh

Ph.D./Thesis

Computer Science

Supervised by

Asst. Prof.

Dr.AbdulSattar M. Khidhir

Asst. Prof.

Dr.Bassam A. Mustafa

Abstract

Most of the earlier attempts of process migration focused on load balancing in distributed systems. However, this is not the only reason for migration. Increasing mobility, application concurrency, efficient resource allocation, distributed systems management, and fault tolerance are some other reasons gained from process migration.

Process migration is the movement of a process from one computing node to another in a distributed environment. Migration requires suspending a process, checkpointing process information, transferring states, and then resuming the transferred process from the point it leaves off.

The objective of the present work is to design and implement a user-level process migration mechanism that exploits the properties of the process's native code and analyzes the binary code in order to achieve migration mechanism that does not rely on Win32 API interception mechanism, nor it alters the operating system kernel.

The current work includes building a system of two parts; one for the source machine at which checkpointing mechanism is done, and the second one for the destination machine where the process would resume execution. The applied migration algorithm is the Total Copy algorithm.

A comparison is made between two types of processes: Win32/CPU bound and Common Language Runtime (CLR) with I/O calls. The obtained time values for checkpointing and address space restoring are approximately twice for CLR process than that for Win32 process, whereas the time values for suspension and context restoring are approximately equal for both process types. The test is achieved in a homogenous environment of two machines running Windows 7 operating system.



جامعة الموصل
كلية علوم الحاسوب والرياضيات

تصميم وتنفيذ هجرة العملية ضمن مستوى الترميز الثنائي

نادية طارق صالح

أطروحة دكتوراه
علوم الحاسوب

بإشراف

أ.م.د. عبد الستار محمد خضر أ.م.د. بسام علي مصطفى

الخلاصة

كانت أغلب المحاولات الأولى لهجرة العملية قد أُلقت الضوء على عملية موازنة الأحمال في النظم الموزعة. لكن هذا الأمر لا يُعد السبب الوحيد للهجرة، حيث أن هناك فوائد كثيرة مُكتسبة من هجرة العملية، منها: زيادة التَّنقل، تزامن التطبيقات، الوصول إلى الموارد بشكل كفاء، إدارة النُظم الموزعة، ومقدار التسامح مع الخطأ.

هجرة العملية هي نقل العملية من حاسوب إلى آخر أثناء وقت تنفيذ النظام في البيئة الموزعة. تتطلب الهجرة الإيقاف الوقتي لتنفيذ العملية، تدقيق وخرن معلومات العملية، نقل معلومات الحالة للعملية، ومن ثم استئناف تنفيذ العملية المنقولة من النقطة التي توقفت عندها.

إن الهدف من هذا العمل هو تصميم وتنفيذ آلية هجرة العملية عند مستوى المستخدم واستغلال خصائص الترميز الأصلي للعملية وتحليل الترميز الثنائي لها من أجل إنجاز آلية هجرة لا تستند على اعتراض دوال واجهة تطبيق البرنامج Win32 API ولا تقوم بتغيير نواة نظام التشغيل. تتضمن هذه الأطروحة بناء نظام يتكون من جزئين، أحدهما خاص بالحاسوب المصدر حيث تُجرى آلية التدقيق والخرن، والآخر خاص بالحاسوب الهدف حيث تُستأنف العملية تنفيذها. وقد تم تطبيق خوارزمية Total Copy في تمثيل آلية الهجرة.

أُجريت مجموعة من المقارنات بين نوعين من العمليات: العمليات المُقَيَّدة بوحدة المعالجة المركزية Win32/CPU bound والعمليات نوع CLR التي تحتوي على استدعاء دوال إدخال/إخراج. ولوحظ أن قيم الوقت التي تم الحصول عليها قد تضاعفت تقريباً في عملية الخرن واسترجاع معلومات الذاكرة في حالة كون العملية من نوع CLR. بينما تكون قيم الوقت متقاربة في عملية الإيقاف الوقتي واسترجاع السياق لكلا النوعين من العمليات. تم تنفيذ العمل في بيئة مُتجانسة مكونة من حاسوبين يعملان تحت نظام التشغيل ويندوز 7.