



البرمجة التفرعية

Parallel Programming

Dr.-Eng. Samer Sulaiman

2023-2024

- أساسيات البرمجة التفرعية
 - مقدمة
 - معامل التسريع
 - أنواع الأنظمة المتعددة المعالجات والبرمجيات الداعمة لها
 - موازنة الأعباء وتحمل الخلل
 - تطبيقات البرمجة التفرعية
 - أشكال معالجة المعطيات على التوازي
- الحواسيب التفرعية
 - تصنيف فلاين Flynn's Classification Scheme
 - شبكات الربط الداخلية Interconnection Networks
- مبادئ تصميم الخوارزميات المتوازية
 - مفاهيم أساسية
 - الإجراءات والمقابلة
 - تقنيات التقسيم
- البرمجيات الداعمة للبرمجة التفرعية
 - المعتمدة على الذاكرة المشتركة
 - المعتمدة على تمرير الرسائل
- تحليل الأداء Performance Analysis

أساسيات البرمجة التفرعية



• مقدمة:

- نظراً لتزايد الحاجة إلى القدرات الحسابية يسعى مصممو الحواسيب باستمرار إلى رفع أداء بُنى حواسيهم.
- تحتاج بعض التطبيقات إلى سرعة حسابية فائقة مثل المحاكاة الرقمية للمسائل العلمية والهندسية، حيث تتطلب غالباً إجراء حساباتٍ تكرارية على حجم هائل من المعطيات قبل الحصول على النتائج المرجوة، والتي ومن المفترض أن تنتهي خلال زمن «معقول» نسبياً.
- بعض المجالات تبدي تحدياتٍ كبيرة في مجال الحسابات مثل نمذجة التراكيب الكبيرة للأحماض النووية والتنبؤ بالأحوال الجوية.
- وتُعدّ مسألة ما ذات تحدٍ كبيرٍ بالنسبة إلى الحسابات إذا كان إنجازها غير ممكن خلال زمن معقول باستخدام الحواسيب المتاحة.
- تُجري الحواسيب التقليدية العمليات المطلوبة بالبرنامج على معالجٍ وحيد،
- لزيادة القدرة الحسابية يتم استخدام حواسيب تتضمن عدة معالجات؛ أو عدة حواسيب تعمل بالتوازي لحل مسألة معرفة،
- وترتبط المعالجات في الحالة الأولى أو الحواسيب في الحالة الثانية بعضها ببعض بطريقة محدّدة لتشكل المنصّة الحسابية أو الحاسوب التفرعي الذي يسمى «نظام متعدد المعالجات».
- في كلا الحالتين تُقسم المسألة المستهدفة إلى أجزاء يجري تنفيذها تفرعياً، بحيث يُنفَّذ كلُّ منها على معالجٍ منفرد.
- تُسمّى كتابة البرامج لنظام متعدد المعالجات بـ «البرمجة التفرعية» أو parallel programming أو «الحوسبة التفرعية».

أساسيات البرمجة التفرعية



• إمكانات زيادة المقدّرات الحسابية

- إن ما يهم بالدرجة الأولى عند تطوير خوارزمية معينة لتنفيذها على نظام متعدد المعالجات؛ هو مدى تسريع عملية الحساب الذي يمكن أن يقدمه هذا النظام مقارنةً بتنفيذ الخوارزمية على نظام وحيد المعالج.
- يمكن حساب معامل التسريع $S(p)$ وفق العلاقة التالية حيث p عدد المعالجات:

$$S(P) = \frac{\text{زمن تنفيذ الخوارزمية على معالج وحيد}}{\text{زمن تنفيذ الخوارزمية على نظام متعدد المعالجات باستخدام } P \text{ معالج}}$$

- إن معامل التسريع الأعظمي النظري وفق العلاقة السابقة هو p بفرض أن زمن تنفيذ الخوارزمية قابل للتقسيم والتوزيع على نحو متساوٍ بين المعالجات،
- تفريع الخوارزمية لا يتطلب أعباءً زمنيةً إضافية.
- لا يمكن الوصول عملياً إلى هذا الحدّ النظري للتسريع؛ إذ تُسهم عوامل متعددة في إضافة زمن في النسخة التفرعية للخوارزمية غير موجود في نسختها التسلسلية يحدُّ من مقدار التسريع

أساسيات البرمجة التفرعية



• العوامل التي تحد من معامل التسريع:

- الفترات الزمنية التي لا يمكن لكل المعالجات أن تنفذ فيها جزءاً من البرنامج،
 - بهذه الحالة تكون هذه المعالجات بوضعية خمول idle وتوقف عن العمل.

- توفر حسابات إضافية في النسخة التفرعية للبرنامج غير ظاهرة في نسخها التسلسلية
 - مثل إعادة حساب بعض المعاملات محلياً.

- الزمن اللازم لتبادل المعطيات بين المعالجات.

- إضافة إلى ذلك فمن المنطقي ألا تقبل بعض مراحل الحسابات التجزئة إلى مهام تفرعية، إذ ينبغي تنفيذها تتابعياً.
 - مثال:

- إذا كان المطلوب هو تفرع خوارزمية ما على p معالجاً، وكان الجزء من الحسابات الذي لا يمكن تجزئته إلى مهام تفرعية هو f ، وبإهمال أي زمن إضافي ناتج من تفرع البرنامج؛ فإن معامل تسريع البرنامج يُعطى بالعلاقة التالية والتي تُعرف بقانون «أمدال»: Amdahl's law:

$$S(p) = \frac{p}{1 + (p-1)f}$$

- وبالتالي فإن معامل التسريع الأعظمي هو: $S(p)_{p \rightarrow \infty} = \frac{1}{f}$

- أي بافتراض وجود 5% من الحسابات ستُجرى تتابعياً في برنامج ما؛ فإن مقدار التسريع الأعظمي هو 20 مرة بغض النظر عن زيادة عدد المعالجات.

أساسيات البرمجة التفرعية



• أنواع الأنظمة المتعددة المعالجات

• يمكن أن تصنّف الأنظمة المتعددة المعالجات من وجهة نظر توزيع الذاكرة في نوعين:

- نظام متعدد المعالجات ذي ذاكرة مشتركة shared memory؛ وهي الأكثر استخداماً في الأنظمة الحديثة،
- نظام متعدد المعالجات ذي ذاكرة موزّعة distributed memory .
- طهر حديثاً صنف جديد يستخدم مجموعة من الحواسيب الشخصية أو محطات العمل لتشكيل منصّة عمل تسمى العنقود. cluster.

• البرمجيات الداعمة للأنظمة المتعددة المعالجات

• جرى تصميم لغات البرمجة التفرعية والبيئات الموافقة تبعاً للنموذج العتادي للنظام التفرعي؛ وتحديداً وفق هيكلية الذاكرة:

- ذاكرة موزّعة
- ذاكرة مشتركة
- نظام عنقودي.

أساسيات البرمجة التفرعية



- الأنظمة المتعددة المعالجات ذات الذاكرة المشتركة
 - تدعم بيئة تعدد المعالجات المفتوحة (Open Multi-Processing (OpenMP) البرمجة بلغة C أو C++ للأنظمة التفرعية ذات الذاكرة المشتركة بهيكلية ونظم تشغيل مختلفة.
 - تقدّم هذه البيئة للمبرمجين واجهات عمل لتطوير التطبيقات التفرعية؛
 - إن كانت على الحواسيب الشخصية المتضمّنة معالجات متعددة النوى أو الحواسيب المخصّصة العالية الأداء.
 - تعتمد OpenMP على تعدد النياسب multithreading، وهي طريقة تفريع parallelizing من خلالها يقوم نيسب thread رئيسي (مجموعة من التعليمات التي تُنفذ تسلسلياً) بالتشعب إلى عدد محدد من النياسب التابعة slaves
 - يجري تقسيم المهمة فيما بينها، بحيث تُنفذ تفرعياً كل منها على نواة أو معالج.
 - يجري تشكيل النياسب للجزء من البرنامج المعدّ للتنفيذ التفرعي قبل التنفيذ.
 - يعطى كل نيسب رقماً تعريفياً خاصاً به، في حين يأخذ النيسب الرئيسي الرقم التعريفي صفراً.
 - بعد التنفيذ التفرعي لهذا الجزء من البرنامج، تعود النياسب التابعة لتنضم إلى النيسب الرئيسي الذي يكمل تنفيذ البرنامج.
 - تتكون الوحدات الأساسية لـ OpenMP من العناصر التالية:
 - توليد النياسب
 - توزيع الأعباء
 - التزامن بين النياسب
 - إدارة معطيات البرنامج ومتحولاته.
- بما أن OpenMP خاصة بالأنظمة ذات الذاكرة المشتركة، فإن متحولات البرنامج العامة تكون متاحة لجميع النياسب، إضافة إلى المتحولات الداخلية الخاصة بكل نيسب.
- توفر مكتبات مثل Pthreads و Shmem العديد من البرامج الفرعية التي تتيح إمكان توليد النياسب وإدارتها، وهذا يؤدي إلى سهولة نسبية بالبرمجة.

أساسيات البرمجة التفرعية



- الأنظمة المتعددة المعالجات ذات الذاكرة الموزعة
 - توفر الواجهة البيئية لتمرير الرسائل Message Passing Interface (MPI) مجموعة من البرامج الفرعية ضمن مكتبة معيارية تُتيح كتابة برامج فعالة بلغات برمجة - مثل C, C++, Fortran- لتبادل الرسائل بين عقد النظام التفرعي وتزامن المهام الموزعة بينها الذي يعتمد على بنية الذاكرة الموزعة،
 - مثل: الحواسيب التفرعية، والعناقيد، والشبكات غير المتجانسة.
 - اعتماداً على برامج المكتبة السابقة يمكن بناء برامج للتراسل في تلك الأنظمة التفرعية.
 - لا تُعدّ MPI لغة برمجة، وإنما يجري استدعاء برامجها الفرعية من قبل برامج النظام.
 - جرى تصميم MPI لتوفر طوبولوجيا افتراضية virtual، والتزامن، وآليات التراسل بين مجموعة من الإجراءات على نحو مستقل عن لغة البرمجة،
 - يجري مقابلة كل إجراء بعقدة معالجة (حاسب أو مخدم)،
 - للحصول على أداء مرتفع يجري عادة نسب إجراءات واحدة إلى كل عقدة معالجة.
 - تدعم جميع منصات عمل أنظمة الحسبات العالية الأداء الواجهة MPI مع قابلية للتوسع scalability.
 - تحتوي الهيكلية الأساسية لبرنامج يستخدم MPI على رابط communicator، وهو متحول variable يُعرّف الإجراءات المسموح بها بالتخاطب فيما بينها،
 - بالإضافة إلى ذلك تحتوي الهيكلية أيضاً على تهيئة لبيئة MPI قبل أن يجري استدعاء أي تابع من توابعها، وإغلاقها عند الانتهاء من تلك البيئة.
 - يشمل تراسل المعطيات في بيئة الـ MPI نوعين:
 - تراسل نقطة إلى نقطة، وفيه يتشارك معالجان فقط بتبادل المعطيات،
 - تراسل المعطيات الجمعي، وفيه تتشارك كل المعالجات المعرفة داخل رابط بتبادل المعطيات.

أساسيات البرمجة التفرعية



• الأنظمة العنقودية

- توفر الآلة الافتراضية التفرعية (PVM) بيئة للمبرمجين لتطوير تطبيقات على أنظمة متعددة المعالجات - تحديداً الأنظمة العنقودية - والتحكم في المهام الموزعة على العقد والتزامن بينها.
- بالإضافة إلى ذلك تتيح تلك البيئة الأدوات البرمجية اللازمة لتبادل الرسائل بين عقد النظام التفرعي.
- تُستخدم هذه البيئة على نطاقٍ واسع؛ إذ تضمن التشغيل البيئي interoperability بين عقد النظام حتى غير المتجانسة وبوجود نظم تشغيل مختلفة.

• نظم التشغيل التفرعية

- تعتمد أنظمة التشغيل الحديثة على منحيين لتلبية الاهتمامات المتزايدة للتطبيقات التفرعية.
- الأول اتخذته نظم التشغيل الحديثة من UNIX وWindows بدءاً من الإصدار NT والتي تهتم بالأنظمة المتعددة المعالجات المتناظرة symmetric multiprocessing،
- الثاني اتخذته نوى نظم تشغيل مخصصة. custom kernels.
- تهتم نظم التشغيل التفرعية بصورة أساسية بالجدولة، والتزامن، وتعدد النيايب، وإدارة الذاكرة وتحمل الخلل.
- يسهل تصميم نظام التشغيل في الأنظمة المتعددة المعالجات المتناظرة؛ إذ يشابه مثيله على معالج وحيد،
- يقوم كل معالج بتنفيذ نسخة من نظام التشغيل؛ وهذا يتيح تنفيذ عدة إجراءات في الوقت ذاته من دون تدني الأداء،
- في حين يصعب تصميم نظام تشغيل للأنظمة المتعددة المعالجات غير المتناظرة.