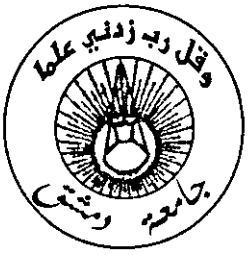
دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	خوارزميات المعالجة المتوازية وبرمجتها
المؤلف الرئيسي:	زين العابدين، كندة
مؤلفين آخرين:	الرجوب، حامد خالد، الخنيفس، خالد علي(مشرف)
التاريخ الميلادي:	2006
موقع:	دمشق
الصفحات:	1 - 172
رقم MD:	575016
نوع المحتوى:	رسائل جامعية
اللغة:	Arabic
الدرجة العلمية:	رسالة ماجستير
الجامعة:	جامعة دمشق
الكلية:	كلية الدراسات العليا
الدولة:	سوريا
قواعد المعلومات:	Dissertations
مواضيع:	الخوارزميات، الحاسبات الإلكترونية ، لغات البرمجة ، شبكات المعلومات
رابط:	http://search.mandumah.com/Record/575016

© 2021 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

الجمهورية العربية السورية
وزارة التعليم العالي
جامعة دمشق



خوارزميات المعالجة المتوازية وبرمجتها

دراسة أعدت لنيل شهادة الماجستير في المعلوماتية

إعداد
كندة زين العابدين

المشرف الأساسي
الأستاذ الدكتور
خالد علي الخنيفس

المشرف المساعد
الدكتور
حامد خالد الرجوب

٢٠٠٦/٩/١٣
٢١ شعبان ١٤٢٧

﴿ بسم الله الرحمن الرحيم ﴾

* وسارعوا إلى مغفرة من ربكم وجنة
عرضها السماوات والأرض أعدت للمتقين *

****

*** كلمة شكر ***

لا يسعني في مقدمة هذه الرسالة إلا أن أشكر

الأستاذ الدكتور خالد علي الخنيفس الذي تفضل

مشكوراً بالإشراف على هذا العمل وكان عوناً لي

على إنجازه وإخراجه بهذه الصورة كما أشكر

الدكتور حامد خالد الرجوب الذي ساهم في

الإشراف على هذه الرسالة

****

الفهرس

مقدمة 1

الفصل الأول: دراسة مفهوم الأنظمة متعددة المعالجات وتعريفها وتصنيفها ودراسة أسلوب التواصل فيما بينها 4

1-1 مفهوم البرمجة متعددة المهام 5

1-1-1 تعريف النظم المعتمدة على الحاسوب 5

1-1-2 تعريف تعددية المهام 6

1-2 الأنظمة متعددة المعالجات 6

1-2-1 وصف الأنظمة متعددة المعالجات 6

1- حواسيب SISD 7

2- حواسيب MISD 7

3- حواسيب SIMD 8

(3-1) حواسيب SM SIMD 8

(3-1-1) حواسيب EREW SM SIMD 9

(3-1-2) حواسيب CREW SM SIMD 9

(3-1-3) حواسيب ERCW SM SIMD 9

(3-1-4) حواسيب CRCW SM SIMD 9

(3-2) حواسيب SIMD شبكة اتصالات 11

(3-2-1) خصائص حواسيب SIMD شبكة اتصالات 12

(3-2-2) الشبكات البسيطة لحواسيب SIMD 12

4- حواسيب MIMD 17

1-2-2 وصف الاجهزة المتوازية 18

1-2-3 جهاز Cray-XMP متعدد المعالجات 19

1- البنية العامة لجهاز Cray-XMP 19

2- جهاز Cray-XMP كآلة شعاعية 20

1-2-4 نمذجة الأنظمة متعددة المعالجات 21

1- نماذج الحساب المتوازي 21

2- نظريات الحساب المتوازي 21

3- الحسابات المتوازية والنماذج اللوغاريتمية 22

4- شروط برنشتاين 27
5- تعريف النموذج اللوغاريتمي 28
6- الدارات المنطقية 31
7- تعاريف 31
8- ملاحظات 38
الفصل الثاني: دراسة الخوارزميات المتوازية وأنواعها وتمييز خواصها وتقنيات تصميمها ومقاييس تحليلها وتحديد كفاءتها 40
2-1 مقدمة عن الخوارزميات المتوازية 41
2-2 تعريف الخوارزمية المتوازية 41
2-3 تحليل الخوارزميات المتوازية 42
2-3-1 أولاً: زمن التنفيذ Running time 42
2-3-2 ثانياً: عدد المعالجات 45
2-3-3 ثالثاً: حساب الكلفة ACCOUNT OF COST 46
2-4 تصميم الخوارزميات المتوازية 47
2-5 كفاءة خوارزمية 53
2-5-1 نظرية(2-1) 54
2-5-2 الإجراء SIMD GAUSS JORDAN (A,b,x) 55
2-6 مقارنة بين الخوارزميات التسلسلية والمتوازية 56
2-6-1 مفاهيم أساسية لبناء خوارزمية متوازية بكلفة أمثلية 57
2-7 خواص الخوارزميات المتوازية 57
الفصل الثالث: خوارزميات متوازية لحل بعض المسائل على أجهزة متعددة المعالجات 59
3-1 خوارزمية التحقق من الانتهاء من التوزيع (خوارزمية النشر) 60
3-2 خوارزمية مراقبة الانتهاء 63
3-3 خوارزمية سبر التفرع الشجري (خوارزمية البحث على الشجرة) 64
3-3-1 خوارزمية الاستعلام 65
1.خوارزمية الموضع 68
2. خوارزمية العدد 69
3. خوارزمية أقرب عنصر 69
4. خوارزمية الرتبة 70
3-4 تطبيقات الخوارزميات المتوازية 70

3-4-1 حساب المجاميع البادئة 70
1- خوارزمية الإجراء : SEQUENTIAL SUMS(X,S) 71
2- خوارزمية الإجراء : PARALLEL SUMS(X,S) 71
3-4-2 حساب جميع المجاميع Computing All Sums 72
3-4-3 خوارزمية المجاميع البادئة على شجرة 76
3-4-4 تحليل خوارزمية المجاميع البادئة على شجرة 78
1- خوارزمية المعالج الورقة 78
2- تحليل خوارزمية المعالج الورقة 79
3-4-5 المجاميع البادئة على الشبكة Mesh 79
3-4-6 تحليل خوارزمية المجاميع البادئة على الشبكة 81
3-4-7 مسألة الأعمال المتسلسلة وفق فترات انتهائها 83
1- خوارزمية الإجراء TREE SEQUENCING(J,answer) 84
2- تحليل خوارزمية الإجراء TREE SEQUENCING 85
3-4-8 مسألة حقيبة الظهر THE KNAPSACK PROBLEM 85
1- خوارزمية الإجراء TREE KNAPSACK 86
2- تحليل خوارزمية حقيبة الظهر 87
3- حلول مسألة حقيبة الظهر على الشبكة 87
3-5 3-5 خوارزميات متوازية لأجهزة متعددة المعالجات ذات ذاكرة مشتركة 88
3-5-1 مسألة الاختيار 88
3-5-1-1 الخوارزمية التسلسلية لمسألة الاختيار 91
1- خوارزمية الإجراء SEQUENTIAL SELECT (S, k) 91
2- تحليل خوارزمية الإجراء SEQUENTIAL SELECT 93
3- الخصائص المطلوبة في الخوارزمية المتوازية للاختيار 94
4- خوارزمية الاختيار المتوازي 96
3-5-2 مسألة الدمج 103
3-5-2-1 شبكة الدمج NETWWORK FOR MERGING 103
3-5-2-2 خواص شبكة الدمج (زوج ، فرد) 107
3-5-2-3 الدمج على النموذج CREW 108
1- إجراء الدمج التسلسلي SEQUENTIAL MERGING 109
2- خوارزمية SEQUENTIAL MERGE (A,B,C) 109

3- الدمج المتوازي PARALLEL MERGING 110
4- خوارزمية BINARY SEARCH (S , x , k) 111
5- خوارزمية CREW MERGE (A , B , C) 112
6- ملاحظات 115
7- تحليل خوارزمية CREW MERGE (A , B , C) 115
3-5-2-4 الدمج على النموذج EREW 116
1- خوارزمية MULTIPLE BROADCAST(d(1),d(2),..,d(N)) 118
2- الخوارزمية الأفضل للنموذج EREW 123
3- خوارزمية إيجاد العنصر الأوسط لمتسلسلتين مرتبتين 124
4- خوارزمية الإجراء Tow_ SEQUENCE MEDIAN(A,B,x,y) 125
5- تحليل خوارزمية الإجراء Tow_ SEQUENCE MEDIAN 128
النتائج والمقترحات 129
المصطلحات المستخدمة 132
قائمة الأشكال والجداول 134
المراجع العلمية 136
ملخص باللغة العربية 138
ملخص باللغة الانكليزية Abstract 154
ملحق لبرنامج دمج مصفوفتين 167

خوارزميات

المعالجة المتوازية

وبرمجتها

مقدمة

تطور علم المعلوماتية منذ بدايته في منتصف القرن العشرين حتى بداية هذا القرن بشكل متسارع وكبير وتشعبت فروعه وتعددت مجالاته واختصاصاته. وأصبح دخول هذا العلم ضرورياً في مختلف مجالات الحياة وكافة فروع العلوم الأخرى واقتضى ذلك تصميم الخوارزميات الملائمة لحل المسائل والمشكلات المطروحة إلى تطوير كبير في علم الخوارزميات. واقتضت بنية الحواسيب التي رافق ظهورها ولادة علم المعلوماتية إلى وضع خوارزميات تنفذ تتابعياً بحيث تكون تعليمات البرامج مرتبة ترتيباً دقيقاً ويتم تنفيذها الواحدة تلو الأخرى، إلا أنه مع زيادة حجم المسائل الواجب حلها وتشعب المشكلات الواقعية المطروحة وصعوبتها أصبح من العسير على مثل هذه الخوارزميات الوصول لحل دقيق لتلك المسائل والمشكلات إما لتجاوز قدرات الذاكرات اللازمة لتخزين المعلومات والنتائج المرحلية أو بسبب المدد الزمنية الكبيرة اللازمة للوصول للحل الأمثل وعلاوة على ذلك ازداد إلحاح مستخدمي الحواسيب في طلب قدرات حسابية متنامية وذلك بشكل مستمر الأمر الذي حدا بالعلماء والباحثين للتفكير في إيجاد أساليب جديدة تساعد على تلبية تلك الرغبات سواء في تسريع عملية البحث عن حلول للمسائل المطروحة أو زيادة حجم تلك المسائل.

ومن هنا ولدت مسألة التوازي في البرمجة إلا أنه تبين استحالة تطبيقها على الأجهزة التقليدية التي تعتمد تقنية الكترونية ثابتة ، الأمر الذي استدعى التدخل في بنية الجهاز نفسه والبحث عن بنية جديدة وقد تم في مطلع الثمانينات من القرن العشرين تصميم أول حاسوب متعدد المعالجات دعي CRAY1 الذي احتوى على معالجين يعملان في آن معاً وتوالى بعد ذلك ظهور حواسيب جديدة تعتمد بنى مختلفة مثل DAP, BSP, Illiaclv, Cyber 205, ... وكان القاسم المشترك بين مختلف هذه البنى هو استخدام عدة معالجات في الحاسوب الواحد، وعلى الرغم من الإمكانات الهائلة التي تقدمها هذه الأجهزة سواء في قدرتها الكبيرة في الحساب أو في حجم الذاكرات الكبير جداً، إلا أن ظهورها قد طرح مشكلة أساسية هي عدم المحافظة على التسلسل الزمني لتنفيذ التعليمات والعمليات مما استدعى تطوير خوارزميات دعيت بالمتوازية لتتلاءم مع الواقع الجديد

الذي فرضته هذه الحواسيب فظهرت خوارزميات متوازية لكل من الأنواع العديدة للحواسيب المتوازية.

ونظراً لأهمية هذا الموضوع في التقدم الحاصل في استخدام المعلوماتية بوصفها الأداة المفضلة في كافة المجالات فقد رغبت بالمساهمة في دراسته لعلي أقدم مساهمة آمل أن تكون ذات فائدة على كافة الأصعدة.

ويمكن اعتبار الأهداف الرئيسية لهذا العمل:

1- عرض مفاهيم الخوارزميات المتوازية.

2- تطوير مفهوم الخوارزميات والمعالجة المتوازية وتطوير إمكانية تطبيق الخوارزميات المتوازية في معالجة الأنظمة التطبيقية.

تقع الرسالة في ثلاثة فصول:

- خصص الفصل الأول لدراسة مفهوم الأنظمة متعددة المعالجات وتعريفها وتصنيفها ودراسة أسلوب التواصل فيما بينها، ويتضمن هذا الفصل تعريف تعددية المهام وتعريف الأنظمة متعددة المعالجات ومن ثم وصفها وتصنيفها، وبعد التعرف على أنواع الحواسيب تبعاً للتعليمات وتحكمها بالبيانات نتعرف على أسلوب الاتصال بين وحدات المعالجة في هذه الأنظمة ونتحدث عنها بالتفصيل. كما أننا نقدم أفكاراً ونتائج تشكل أساسات نظرية الحساب المتوازي ونوضح بعض نماذج الحساب المتوازي. كما نتحدث عن الدارات المنطقية التي تعتبر نموذجاً مصغراً عن الحساب المتوازي وتعتبر مقياساً لكلفة التوازي فهي تشكل قاعدة لدراسة تعقيد الحسابات المتوازية.
- خصص الفصل الثاني لدراسة الخوارزميات المتوازية وأنواعها وتمييز خواصها وتقنيات تصميمها ومقاييس تحليلها وتحديد كفاءتها، ويتضمن هذا الفصل مقدمة عن الخوارزميات المتوازية وتعريفاً لها ومن ثم تحليل الخوارزميات المتوازية وتحديد المعايير الخاصة بذلك وبعدها نقوم بشرح بعضاً من تقنيات تصميم الخوارزميات المتوازية والتسلسلية وأخيراً نحدد خواص الخوارزميات المتوازية.
- خصص الفصل الثالث لدراسة خوارزميات متوازية لحل بعض المسائل على أجهزة متعددة المعالجات، ويتضمن هذا الفصل شرحاً مفصلاً لخوارزميات متوازية

على نماذج مختلفة من الحواسيب متعددة المعالجات ومن ثم تحليلها وذكر خواصها ومقارنتها بالخوارزميات التسلسلية للمسألة. إضافة إلى عرض بعض تطبيقات الخوارزميات المتوازية مثل مسألة حقيبة الظهر ومسألة المجاميع البادئة ومسالة الأعمال التسلسلية وفق فترات انتهائها كما نبين فيما بعد الخوارزميات المختلفة لمسائل رياضية شهيرة على أشكال مختلفة من الأجهزة متعددة المعالجات ونذكر الأمثلة التوضيحية الموافقة لكل منها.

كما تحتوي الرسالة على ملحق يتضمن البرامج الحاسوبية لبعض الخوارزميات إضافة لعرض بعض الأمثلة التوضيحية لعملها.

وقد توخيت البساطة في العرض مع المحافظة على الدقة العلمية، كما أني أسقطت إثبات بعض النظريات التي لا مناص من ذكرها حفاظاً على التسلسل المنطقي في عرض المادة العلمية وفهمها.

وإني أرجو الله أن أكون قد وفقت في معالجة هذا الموضوع وقدمت مساهمة بسيطة في تطوير جانباً مهماً من جوانب علم المعلوماتية.

وأسأل الله العون إنه نعم المولى ونعم النصير.

الفصل الأول

دراسة مفهوم الأنظمة متعددة المعالجات وتعريفها وتصنيفها ودراسة أسلوب التواصل فيما بينها

1-1 مفهوم البرمجة متعددة المهام:

1-1-1 تعريف النظم المعتمدة على الحاسوب:

تعرف النظم المعتمدة على الحاسوب بأنها مجموعة أو ترتيب عناصر نُظمت بغية إنجاز هدف محـــدد بمعالجة المعلومات. ويستخدم النظام المعتمد على الحاسوب عدداً من العناصـــر وهـــي: البرمجيـــات والعتاديات والأشخاص وقاعدة المعطيات والوثائق والإجراءات. وتُضم هذه العناصر بأشكال مختلفـــة لتحويل المعلومات.

ولإنشاء نماذج لهذه النظم نقوم بما يلي:

- تعريف الإجرائيات التي تخدم احتياجات الرؤية المعينة
- تمثيل سلوك الإجرائيات والافتراضات التي يستند إليها هذا السلوك
- تعريف الدخل الخارجي والدخل الداخلي للنموذج بشكل واضح
- تمثيل جميع الروابط (بما فيها الخرج) التي تساعد المهندس على فهم الرؤية فهماً أفضل.

ولبناء نموذج نظام يجب على المهندس التقيد بالعوامل التالية:

1- الافتراضات: التي تخفض عدد التبادلات والاختلافات الممكنة والتي تسمح من ثم للنموذج أن يعكس المشكلة على وجه مقبول.
2- التبسيطات: التي تسمح بإنشاء النموذج بالوقت المناسب.
3- الحدود: التي تساعد في وضع حدود للنظام.
4- القيود: التي توجه الطريقة التي يُنشأ بها النموذج والمنهج المتبع في تنجيزه.
5- التفضيلات التي تدل على البنيانات المفضلة لجميع المعطيات والوظائف والتقانة.

قد يترتب على نموذج النظام الناتج حل تام الأتمتة أو نصف مؤتمت أو يدوي. ويمكن وصف نمـــاذج لكل نمط يحقق حلاً بديلاً للمسألة المطروحة وببساطة يتم تعديل التأثير النسبي لعناصر النظام المختلفة (الأشخاص والعتاديات والبرمجيات) بغية استنتاج نماذج من كل نمط.

2-1-1 تعريف تعددية المهام:

تعرف تعددية المهام (Multitasking)على أنها قدرة نظام التشغيل على تنفيذ أكثر من مهمة واحدة في الوقت نفسه وتتيح تعددية المهام مثلاً تنفيذ عدة برامج مختلفة والطباعة على الطابعة والتخزين على القرص في الوقت نفسه ودون انتظار انتهاء كل مهمة من هذه المهمات قبل تنفيذ الأخرى.

وتبنى الخوارزمية في الأجهزة المتعددة المعالجات على أساس تعددية المهام والتنفيذ المتزامن حيث تتولى كل وحدة معالجة مسألة جزئية فتقوم بمعالجتها بشكل متزامن مع الوحدات الأخرى ونحصل في النهاية على النتيجة بجمع تلك النتائج المتعددة والحصول على الحل النهائي للمسألة. وتعتمد البرمجة المتعددة المهام على فهم طريقة تنفيذ التعليمات على البيانات حيث إن هناك سلسلة من التعليمات تأمر الحاسب بوظيفته في كل خطوة وهناك سلسلة من البيانات (دخل الخوارزمية) ستتأثر بتلك التعليمات.

ويتم تصنيف الحواسيب تبعاً لطريقة تنفيذ التعليمات على البيانات وتتم برمجة كل نوع من أنواع الحواسيب باستخدام خوارزمية إما تسلسلية وإما متوازية كل على حسب ما يوفر من إمكانيات. كما أنه من الضروري وجود اتصال بين وحدات المعالجة بعضها مع بعض خلال الحساب وذلك من أجل تغيير البيانات أو النتائج ويتحقق هذا الاتصال عن طريق الذاكرة المشتركة للنظام أو عن طريق شبكة الاتصالات بين وحدات المعالجة.

2-1 الأنظمة المتعددة المعالجات:

1-2-1 وصف الأنظمة المتعددة المعالجات:

لقد وجدت الأنظمة المتعددة المعالجات من أجل تحسين الفعاليات وذلك مقارنة بالكلفة والموثوقية والتكيّف ويعتبر النظام المتعدد المعالجات (مع أو بدون توازٍ) رائداً بين أنظمة المعلوماتية حيث ينافس نظام المعالج الوحيد وينافس أيضاً الأنظمة الموزعة وغيرها.

وفي النظام المتعدد المعالجات المتوازي يوجد العديد من وحدات المعالجة المستقلة والتي تتصف بذاكرات محلية وتشترك معاً بذاكرة رئيسية وتكون هذه المعالجات متزامنة. حيث يحوي الحاسب المتوازي عدة معالجات ويقوم بإعطاء مسألة جزئية من المسألة الكلية لكل منها وتحل المسائل الجزئية بشكل متزامن كل واحدة في معالج مختلف ومن ثمّ تجمع النتائج لإعطاء حل المسألة الأصلية.

يتكون أي حاسب سواء تسلسلي أو متوازٍ من تعليمات تنفذ على البيانات: حيث تـأمر سلسـلة مـن التعليمات الحاسب بما سيقوم به في كل خطوة ، وتتأثر سلسلة بيانات (وهي دخل الخوارزمية) بتلـك التعليمات.

نستطيع أن نميز بين أربعة أنواع من الحواسيب:

1- أولاً: حواسيب SISD (سلسلة تعليمات مفردة، سلسلة بيانات مفردة):

Single Instruction stream, Single Data stream(SISD)

وهي حواسيب تعلمية – تعليمية حيث يتكون الحاسب في هذه الفئة من وحدة معالجـة مفـردة تتلقـى سلسلة تعليمات مفردة. تصدر وحدة التحكم تعليمة واحدة تطبقها على عنصر البيانات الذي تحصـل عليه من وحدة الذاكرة. نستخدم لبرمجة حاسب من هذه الفئة خوارزميـة تسلسـلية أو تتابعيـة.
[SeGA-1989]

٦٣٥٧٥٩

2- ثانياً: حواسيب MISD (سلسلة تعليمات متعددة – سلسلة بيانات مفردة):

Multiple Instruction stream, Single Data stream (MISD)

يوجد في هذا النوع من الحواسيب N معالجاً كل منها يملك وحدة تحكم خاصة به ويتقاسم وحدة الذاكرة المشتركة حيث توجد البيانات. يصل في كل خطوة عنصر واحد من البيانات من الذاكرة ويتم العمل به من قبل جميع المعالجات بآن واحد كلٌ حسب التعليمة التي وصلت إليه من وحدة تحكمـه الخاصـة. يتحقق التوازي هنا بجعل المعالجات تعمل عدة أشياء مختلفة في الوقت نفسه على عنصر البيانات نفسه وهذا النوع من الحواسيب يساعد نفسه بشكل طبيعي من أجل الحسابات التي تتطلب إدخالاً ليصبح مادة لعمليات متعددة فيتلقى كل منها الإدخال بالشكل الأصلي.[SeGA-1989]

مثال1-1:

لتكن لدينا مسألة تحديد ما إذا كان عدد معطى z أولياً أو لا.

إن الحل المعروف لهذه المسألة هو تجريب جميع احتمالات قاسم العدد Z : فإذا لم ينجح أي منها فـي القسمة على Z عندها يكون أولياً وإلا فإنه يكون مركباً أو غير أولي.

تكمن الفكرة هنا في تقسيم العمل في اختبار القواسم الممكنة بين المعالجات. حيث تأخذ كل المعالجات Z على أنه إدخال ومن ثم يحاول كل منها تقسيم العدد على القواسم المحتملة المساعدة الموجودة عنده

ويناقش قاعدة الخرج المخصصة على النتيجة. وهكذا يمكن تحديد ما إذا كان Z أولياً بخطوة واحـدة. أما إذا كان لدينا عدد قليل من المعالجات أقل من القواسم المحتملة فإن كل من المعالجـات يسـتطيع إنجاز العمل في اختبار مجموعة فرعية مختلفة من القواسم.

3- ثالثاً: حواسيب SIMD (سلسلة تعليمات مفردة – سلسلة بيانات متعددة):

Single Instruction stream, Multiple Data stream (SIMD)

يتكون الحاسب المتوازي في هذه الفئة من N معالجاً متماثلاً كل واحد منها يمتلـك ذاكرتـه المحليـة الخاصة والتي يمكن أن يخزن فيها البرنامج والبيانات. تعمل كل المعالجات تحت تحكم سلسلة تعليمات مفردة تمت معالجتها في وحدة التحكم المركزية. وبشكل مكافئ فإنه من المفترض أن تمتلك كل مـن الـ N معالج نسخاً متطابقة من البرنامج المفرد. وتخزن كل نسخة برنامج في الذاكرة المحلية لكـل معالج وهناك أيضاً لكل معالج N سلسلة بيانات. تعمل المعالجات بشكل متزامن ، حيـث تنفـذ كـل المعالجات التعليمة نفسها في كل خطوة ويمكن أن تكون التعليمة بسيطة (مثل جمع أو طرح عددين) أو تعليمة معقدة مركبة (مثل دمج قائمتين من الأرقام) وبشكل مشابه يمكن أن يكون عنصر البيانات بسيطاً (رقماً واحداً) أو مركباً (عدة أرقام).

ربما يكون ضرورياً في بعض الأحيان أن نملك مجموعة فرعية فقط من المعالجات تقوم بتنفيذ تعليمة ما ويمكن أن نرمز هذه المعلومات في التعليمة نفسها فنخبر المعالج ما إذا كان عليه أن يكـون فعـالاً (وينفذ التعليمة) أو غير فعال (وينتظر من أجل أن يتلقى التعليمة التالية).

في معظم المسائل الهامة التي نحلها باستخدام حواسيب من هذا النوع يكون من المسـتحب أن تكـون المعالجات قادرة على الاتصال مع بعضها خلال الحساب من أجل تغيير البيانات أو النتائج المكـررة. ونحقق هذا الاتصال وفقاً لفئتين فرعيتين: إحداهما يكون الاتصال فيها عن طريق الذاكرة المشـتركة والأخرى عن طريق شبكة الاتصالات.

(3-1) حواسيب SIMD (SM) بذاكرة مشتركة Shared-Memory :

تعرف هذه الفئة أيضاً بآلة الوصول العشوائي المتوازي نموذج (PRAM) The Parallel Random Access Machine .

يتم تقاسم الذاكرة المشتركة هنا والتي يستخدمها مجموعة من الناس بنفس الطريقة ، عندما يريد معالجـان الاتصال فإنهما يعملان خلال الذاكرة المشتركة . لنفترض أن المعالج I يرغب بتمرير قيمة للمعالج j يـتم

هذا بخطوتين: الأولى يكتب المعالج I الرقم في قسم الذاكرة في موقع معروف من قبل المعالج j والذي يقرؤه بدوره من ذلك الموقع.

خلال تنفيذ خوارزمية التوازي فإن الـ N معالج تصل إلى الذاكرة المشتركة لقراءة البيانات المدخلة من أجل قراءة أو كتابة النتائج المكررة ومن أجل كتابة النتائج النهائية. يسمح النموذج الأساسي لجميع المعالجات بالوصول إلى الذاكرة المشتركة بنفس الوقت وذلك إذا كانت مواقع الذاكرة المحددة من أجل القراءة منها أو الكتابة فيها مختلفة.

من ناحية أخرى يمكن أن نقسم فئة الحواسيب هذه إلى أربع فئات فرعية وفقاً لقدرة اثنين أو أكثر من المعالجات على الوصول إلى موقع الذاكرة نفسه وبنفس الوقت وذلك كما يلي:

Exclusive-Read, Exclusive-Write (EREW) SM SIMD Computers (1-1-3)

في هذا النوع يكون الوصول إلى مواقع الذاكرة محدوداً بمعنى أنه لا يُسمح لمعالجين القراءة من الموقع نفسه من الذاكرة أو الكتابة في الموقع نفسه من الذاكرة وذلك في الوقت نفسه.

Concurrent-Read, Exclusive-Write (CREW) SM SIMD (2-1-3) Computers.

يسمح هذا النوع من الحواسيب بالقراءة المتعددة للمعالجات من نفس موقع الذاكرة ولكن من أجل الكتابة فإنه يبقى محدوداً أي أنه لا يسمح لمعالجين بالكتابة في نفس موقع الذاكرة بالوقت نفسه.

Exclusive-Read, Concurrent-Write (ERCW) SM SIMD (3-1-3) Computers.

يسمح هذا النوع من الحواسيب بالكتابة المتعددة للمعالجات في نفس موقع الذاكرة ولكن الوصول من أجل القراءة يبقى محدوداً إذ أنه لا يمكن لمعالجين القراءة من نفس الموقع من الذاكرة في الوقت نفسه.

Concurrent-Read, Concurrent-Write (CRCW) SM SIMD (4-1-3) Computers.

في هذا النوع من الحواسيب تتحقق ميزة القراءة المتعددة المتداخلة والكتابة المتعددة للمعالجات المتعددة.

إن خاصية القراءة المتعددة والوصول إلى نفس الموقع من الذاكرة لن يؤدي إلى أية مشاكل حيث أن كل من المعالجات التي تقرأ من ذلك الموقع تصنع نسخة من محتويات الموقع وتخزنها في ذاكرتها المحلية الخاصة ولكن المصاعب تظهر عند الوصول للكتابة المتعددة . فإذا حاولت عدة معالجات أن تخزن بيانات مختلفة في عنوان محدد بنفس الوقت فأي منها سينجح؟ وبمعنى آخر حتماً سيكون هناك طريقة لتحديد محتويات ذلك العنوان بعد عملية الكتابة.

مثال 1-2:

ليكن لدينا ملف كبير جداً يتألف من n إدخال منفصل وسنفترض للسهولة أن الملف غير مرتب بأي طريقة. والمطلوب تحديد فيما إذا كان العنصر x موجود في الملف أم لا من أجل إنجاز عملية قاعدة بيانات قياسية مثل القراءة والتعديل والحذف.

في حواسيب SISD التقليدية مثلاً تكون عملية استعادة x مطلوبة n مرة في أسوأ الحالات حيث يوجد في كل خطوة مقارنة بين x وإدخال الملف. وتكون الحالة الأسوأ عندما يكون x مساوياً لآخر إدخال أو غير مساوي لأي إدخال. ويصبح العمل أقل جودة بشكل متوسط كما يلي: إذا كانت إدخالات الملف موزعة بانتظام على المجال المعطى عندها تُطلَب نصف الخطوات العديدة لاستعادة x وهو العمل الذي نستطيع القيام به بشكل أسرع على حواسيب EREW SM SIMD مع N معالج حيث: N<n.

لنرمز للمعالجات بـ:$p_1,p_2,..,p_n$

نحتاج من أجل جعل جميع المعالجات تعرف قيمة x في البداية إلى استعمال تعليمة النشر أو الإعلان BROUDCASTING كما يلي:

1- p_1 يقرأ x ويبلغ p_2
2- بنفس الوقت p_1,p_2 تبلغ p_3,p_4 على التوالي
3- بنفس الوقت p_1,p_2,p_3,p_4 تبلغ p_5,p_6,p_7,p_8 على التوالي

وهكذا..

تستمر هذه العملية حتى تحصل جميع المعالجات على x حيث يتضاعف عدد المعالجات في كل مرحلة. يتطلب إعلان x إلى log n خطوة.

ليكن الملف الذي نبحث فيه عن x مقسم إلى ملفات جزئية يتم فيها البحث من قبل المعالجات بنفس الوقت: p_1 يبحث في العنصر n/N الأول وهكذا..

بما أن جميع الملفات الجزئية لها نفس الحجم فإننا سنحتاج إلى n/N خطوة في أسوأ الأحوال للإجابة عن استفسار x وبالنتيجة ومن أجل ذلك تتطلب الخوارزمية المتوازية log N + n/N خطوة في أسوأ الأحوال. لنأخذ موقع f يملك قيمة منطقية يحدد جانباً في الذاكرة المشتركة ويشير إلى أن أحد المعالجات قد وجد العنصر المطلوب وبالتالي فإن جميع المعالجات الأخرى تنهي عملها. مبدئياً تأخذ f القيمة false وعندما يجد المعالج x في ملفه الجزئي يضع في f قيمة true.

في كل خطوة من البحث تفحص جميع المعالجات قيمة f فإذا كانت true تتوقف.

نلاحظ أن هذا التعديل لم يأت بالسهولة فنحن بحاجة إلى log N خطوة للإعلان عن قيمة f في كل مرة تحتاجها المعالجات. أي إلى log N + n/N log N خطوة في أسوأ الأحوال. ولكي نستغل هذه النتيجة بدون زيادة زمن التنفيذ فإننا بحاجة إلى استعمال نماذج أكثر قوة تدعى حواسيب CREW SM SIMD. بما أن عمليات القراءة المتعددة مسموحة فإنها تأخذ خطوة واحدة لجميع المعالجات لكي تحصل على x وخطوة واحدة لتقرأ f في كل مرة تحتاجها. وهذا يقودنا إلى أن الحالة الأسوأ تأخذ n/N خطوة.

ربما يمثل الملف قاعدة بيانات نصية مع مئات الآلاف من البنود والفقرات كل منها يشمل عدة آلاف من الكلمات، وربما يكون ضرورياً البحث في مثل هذا الملف عن كلمة محددة x وهنا فإننا قد نجد أكثر من إدخال واحد مساوي لقيمة x ومن ثم أكثر من معالج يحتاج إبلاغ النجاح في نفس الوقت وهذا يعني أن اثنين أو أكثر من المعالجات ستحاول الكتابة في الموقع f في الوقت نفسه وهذا الشيء ممكن فقط في حاسبCRCW SM SIMD.

(3-2) حواسيب SIMD شبكة اتصالات:

يمكننا الحصول على نموذج مرن أكثر قوة من نموذج الذاكرة المشتركة يكون فيه كل زوج من المعالجات متصل بخط ثنائي الاتجاه ، حيث يمكن لعدة أزوج أن تتصل بآن واحد (ولكن بشرط ألا يحاول أكثر من معالج إرسال البيانات إلى معالج آخر أو تلقي البيانات من معالج آخر).
وبالتالي فإنه من المحتمل أن تستطيع جميع المعالجات أن تكون مشغولة بالاتصال جميع الوقت وهذا غير ممكن في ذاكرة مشتركة مقسمة إلى مقاطع R-block shared memory عندما يكون عدد المعالجات N أكبر من عدد المقاطع R .

(3-2-1) خصائص حواسيب SIMD شبكة اتصالات:

1- **الكلفة (السعر):** إذا كان السؤال المطروح : ما هو السعر المدفوع من أجل شبكة اتصالات تامة بـــ N معالج ؟ هناك (N-1) خط مسموح لكل معالج ومنه لدينا بشكل كامل N(N-1)/2 خـــط. من الواضح أن مثل هذه الشبكة يكون سعرها غالي جداً وبشكل خاص من أجل قيمة كبيرة لـــــ N. وهذا صحيح على الأخص إذا لاحظنا أنه مع N معالج فإن أفضل ما يمكن من أجل N ملف تخفيض عدد الخطوات المطلوبة في الخوارزمية التسلسلية.

2- **الوثوقية Feasibility:**

حتى لو استطعنا تحمل مثل هذا السعر العالي فإن النموذج غير واقعي عملياً مرة ثانية من أجل قيمـــة كبيرة لـــ N. بالإضافة إلى أنه هناك حد لعدد الخطوط التي يمكن وصلها للمعالج وهذا الحد يفرضـــه الحجم الفيزيائي الحقيقي للمعالج نفسه.

3- **العلاقة بالنموذج SM SIMD:**

أخيراً من الملاحظ أن نموذج الاتصالات التام[1] أضعف من حاسب بذاكرة مشتركة من أجـــل نفـــس السبب . مثلاً في الذاكرة المشتركة المقسمة إلى R مقطع لا يستطيع أكثر من معالج واحد الوصـــول المحدد بآن واحد إلى مقطع الذاكرة بالمساعدة مع معالج آخر. وهذا يؤدي إلى كلفة مماثلة تقريباً لكلفة حاسب SM SIMD (ولكن دون أن نحسب الكلفة من الدرجة الثانية لطرق ثنائية الاتجاه). وهذا يحبط غرضنا الأصلي في الحصول على آلة عملية أكثر.

(3-2-2) الشبكات البسيطة لحواسيب SIMD:

من حسن الحظ أنه في معظم التطبيقات هناك مجموعة جزئيـــة صـــغيرة مـــن جميـــع الارتباطـــات (الاتصالات) المزدوجة تكفي غالباً للحصول على أداء جيد.

سنلخص بإيجاز الشبكات الأكثر شيوعاً منها فيما يلي:

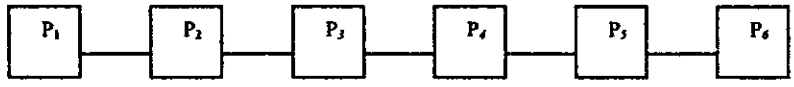
لنفترض أن لدينا معالجين يستطيعان الاتصال بعدد ثابت من الخطوات في حاسب SM SIMD . إن أي حاسب SIMD بشبكة اتصالات يمكن أن يُحاكى على نموذج سابق بدون أن يحتاج خطوات أكثر من الخطوات المطلوبة لإيجادها كما في السابق .

---

[1] نموذج الاتصالات التام: تتوزع في هذا النموذج مواقع الذاكرة المشتركة الـــ M بين N معالجاً ، كل منهـــا يتلقـــى M/N موقعاً. إضافة إلى أن كل زوج من المعالجات تتصل فيما بينها بخط ثنائي الاتجاه.

1- مصفوفة خطية Linear Array:

يمكن تمثيل وصل N معالج بشكل مصفوفة وحيدة البعد كما يظهر في الشكل (1-1) من أجل N=6.
هنا المعالج P_i مرتبط بمجاوريه P_{i-1} و P_{i+1} من خلال خط اتصال ثنائي الاتجاه.
كل من معالجات النهاية والمسماة P_1 و P_N تملك فقط مجاور واحد.



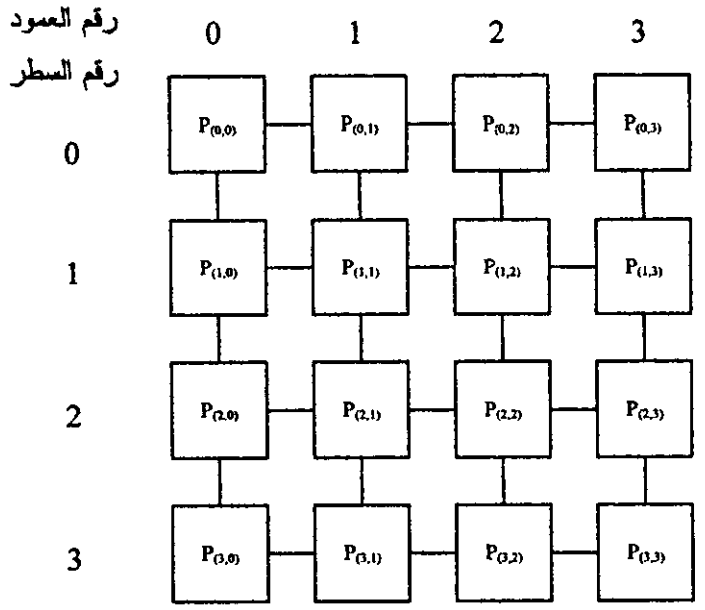
شكل (1-1) شبكة خطية من أجل N=6

2- مصفوفة ثنائية البعد Tow-Dimensional Array:

نحصل على شبكة ثنائية البعد بترتيب الـ N معالج في مصفوفة m×m حيث $m = N^{1/2}$ كما يظهر الشكل(2-1) من أجل m=4: المعالج في السطر j والعمود k يرمز بـ $p_{(j,k)}$ حيث: $0 \le k \le m-1$ و $0 \le j \le m-1$. هناك خط ثنائي الاتجاه يربط $p_{(j,k)}$ بمجاوريه $p_{(j-1,k)}$ و$p_{(j+1,k)}$ و$p_{(j,k+1)}$ $p_{(j,k-1)}$. تملك المعالجات في الحدود الفاصلة للأسطر و الأعمدة عدد اقل من أربع مجاورات وبالتالي اتصالات أقل.

تعرف هذه الشبكة أيضاً بـ the mesh.

نلاحظ أنه في كل من المصفوفتين أحادية البعد وثنائية البعد تملك الشبكة خاصية مهمة: وهي أن جميع الخطوط في الشبكة تملك نفس الطول وهي غير موجودة في الاتصالات الأخرى التي سندرسها.



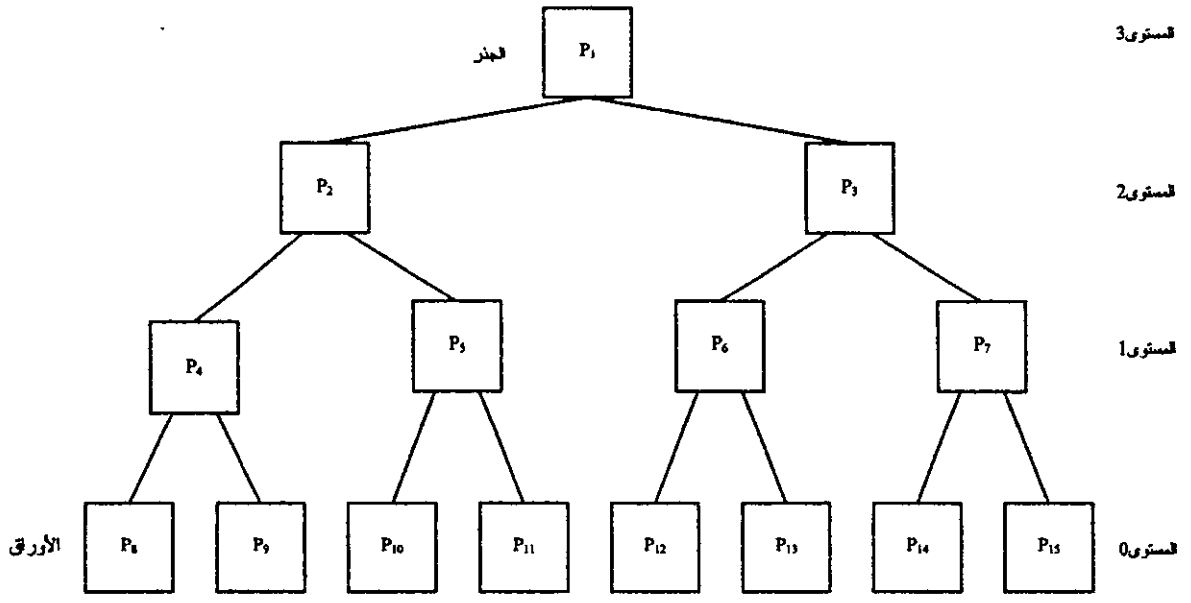
شكل1-2 شبكة اتصالات بمصفوفة ثنائية البعد (شبكية)

من أجل m=4

3- شجرة الاتصال Tree Connection

تكون المعالجات في هذه الشبكة بشكل شجرة ثنائية تامة وتمثل هذه الشجرة تملك d مستوى مرقمة من 0 إلى d-1 و $N=2^d-1$ وينتهي كل من تلك بمعالج كما يظهر الشكل (1-3) من أجل d=4.

كل معالج في المستوى i يتصل بخط ثنائي الاتجاه بسلفه في المستوى i+1 ويتصل بخلفه في المستوى i-1 . المعالج الجذر في المستوى d-1 لا يملك سلفاً وكل من الأوراق في المستوى(0) لا يملك خلفاً أو أوراقاً.



شكل (1-3) شبكة اتصالات شجرية من أجل d=4

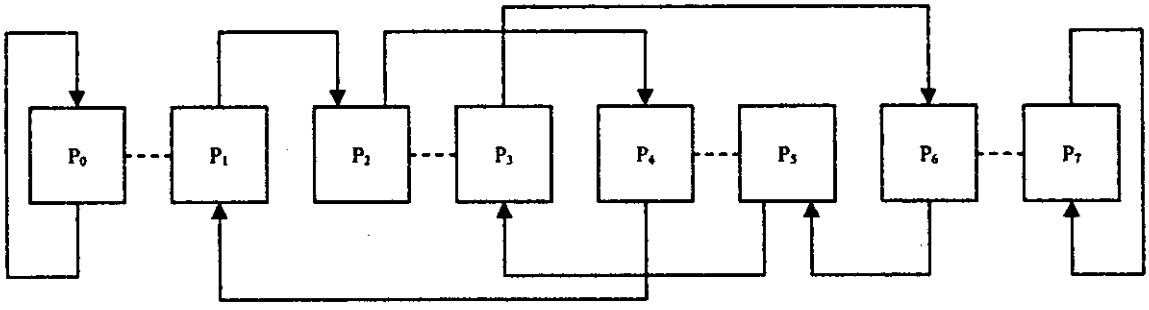
4- الاتصالات المختلطة التامة Perfect shuffle connection:

ليكن لدينا N معالج $P_0,P_1,P_2,..,P_{N-1}$ متوفرة حيث N هي قوة لـــ 2 :($N=2^x$)

يوجد في الاتصالات المختلطة التامة خط وحيد الاتجاه يربط P_i بـــ P_j حيث:

$$J=\begin{cases} 2i & \text{من أجل } 0 <= I <= N/2 - 1 , \\ 2i + 1 - N & \text{من أجل } N/2 <= I <= N-1 , \end{cases}$$

كما يُظهر الشكل(1-4) من أجل N=8



شكل (1-4) شبكة اتصالات مختلطة تامة من أجل N=8

بشكل مكافئ نحصل على التمثيل الثنائي لـ j بتغيير دوري لـ i موضع واحد إلى اليسار ويوضح ذلك وفق ما يلي في الجدول 1-1.

p_i المعالج	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
p_j المعالج	P_0	P_2	P_4	P_6	P_1	P_3	P_5	P_7

جدول 1-1 يبين الاتصالات لشبكة مختلطة تامة من أجل N=8

بالإضافة إلى أنه في بعض الأحيان تضاف إلى الشبكة خطوط ثنائية الاتجاه تصل كل معالج إلى الذي خلفه بشكل دوري على شكل ثنائيات في المثال:

(0,1)،(2,3)،(4,5)،(6,7) ثم (1,2)،(3,4)،(5,6)

وتدعى هذه الارتباطات بالارتباطات المتغيرة، وهي تظهر كخطوط مقطعة في الشكل(1-4). وفي هذه الحالة تعرف الشبكة بالارتباطات المتغيرة المختلطة.

5- الاتصالات التكعيبية Cube Connection:

لنفترض أن $N=2^q$ من أجل بعض $q\geq 1$ ولتكن الـ N معالج: $P_0,P_1,P_2,..,P_{N-1}$ مكعب من q بعد أو (hypercube) نحصل عليه بوصل كل معالج إلى q مجاور.

الـ q مجاور p_j من p_i معرفة كما يلي: نحصل على التمثيل الثنائي لـ j بتكملة البت المفرد في i. وهذا موضح في الجدول (1-2) لقيم i وj دليلي المعالجين المتجاورين حيث q=3 والفهرس لـ $p_0,p_1,..,p_7$ معطى في مجموعة رموز ثنائية.