تصميم وتنفيذ خوارزميات المعالجة المتوازية

**وسام علي حسين سلمان الخزعلي**

**مدرس مساعد**

**ماجستير علوم الحاسوب وتقنية المعلومات**

**الملخص**

تعتبر تقنية المعالجة المتوازية فرع جديد وهام من فروع المعلوماتية لما تظمنته من افاق عظيمة وكبيرة لتوسيع وتسريع عمليات المعالجة في الكمبيوتر، ولضمان عمل هذه التقنية بصورة صحيحة وفعالة وجدت عدة طرق لادارة هذه العمليات تدعى بالخوارزميات (خوارزميات المعالجة المتوازية) ونظرا لاهمية هذه التقنية وحداثتها قمنا بالولوج في هذا العلم من خلال بحثنا هذا، فقد ناقشنا في الباب الاول ما هي المعالجة المتوازية وما هي انواع حواسيبها اضافة الى التعريف بخوارزميات المعالجة المتوازية وكيفية تحليلها، كما ناقشنا في الباب الثاني البحوث والدراسات التي تناولت هذه التقنية، اما في الباب الثالث فقد تناولنا كيفية تصميم وبناء خوارزميات المعالجة المتوازية وكمثال على ذلك قمنا ببناء خوارزمية تدعى بخوارزمية النشر والتي تمكن من وصول اكثر من معالج واحد الى موقع معين في الذاكرة في نفس الوقت، وختمنا بحثنا هذا ببعض النتائج المتحققة من هذا البحث اضافة الى بعض التوصيات ومن الله التوفيق.

**Abstract.**

Parallel processing technology is considered a new and important branch of the IT branch of what it contained a great big prospects to expand and accelerate the processing in the computer, and to ensure the work of this technique correctly and effectively found several ways to manage these processes, called algorithms (parallel processing algorithms) Because of this technology, and timeliness of the importance we Login to this science through this our research, we discussed in the first section what is the parallel processing and what are the types of its computers in addition to the definition of parallel processing algorithms and how to analyze. As we discussed in Part II of research and studies on this technique, either in Part III we have considered how to design and build processing algorithms parallel As an example, we have built an algorithm called algorithm publication and which enables the arrival of more than one processor to a specific location in memory at the same time. The conclusion was some of the results obtained from this research as well as some of the recommendations are reconciled to God.

**مقدمة**

منذ بداية عصر معالجة المعلومات ادرك العالم ان من المفيد جعل الاجزاء المختلفة للحاسوب تقوم بتنفيذ اعمال مختلفة في الوقت نفسه، بينما تقوم وحدة المعالجة المركزية بالعمليات الحسابية بأمكانها ايضا ان تقرا المدخلات من وحدات الادخال وتخرج المخرجات على وسائل الاخراج المتاحة.

حيث تحتوي الاجهزة الاكثر تقدما على عدة معالجات يختص كل منها بعمل معين وتشترك فيما بينها لانجاز المهام المطلوب معالجتها. وبالرغم من ان سرعة المعالجات قد ازدادت بشكل كبير الا ان الطلب على معالجة اسرع للمهام اصبح ضروريا بشكل اكبر، وقد قاد ظهور اجهزة المعالجة المتوازية ذات المعالجات المتعددة (المتزامنة وغير المتزامنة) وظهور الشبكات الكبيرة الى اعادة التفكير في مفهوم الخوارزميات، اذ ان الخوارزمية تهدف بعد ان تتحقق من وجود الحل الى البحث عن حل عملي، كما ان الكمبيوتر المتوازي الذي يملك العديد من وحدات المعالجة يقوم بتجزئة المسألة المعطاة الى مسائل جزئية يقوم بحلها بشكل متزامن كل مسألة جزئية بمعالج مختلف، وبعد ذلك تجمع النتائج لاعطاء حل للمسألة الاصلية. ومع ظهور الحواسيب المتوازية اصبح من الضروري البحث عن كيفية برمجتها لحل المسائل بشكل فعال وعملي وبطريقة ذات كلف منخفضة، حيث تسمى تلك الطرق بالخوارزميات (خوارزميات البرمجة المتوازية).[1]

**2.1 المعالجة المتعددة المهام**

هو قدرة الكمبيوتر على تنفيذ اكثر من مهمة واحدة في نفس الوقت، حيث تتيح تعددية المهام تنفيذ عدة برامج مختلفة اضافة الى اجراء عملية الطباعة على الطابعة واجراء عمليات التخزين على القرص، دون انتظار انتهاء كل مهمة من تلك المهمات للبدء بتنفيذ مهمة اخرى.[2]

**3.1 الانظمة المتعددة المعالجات**

وجدت الانظمة المتعددة المعالجات من اجل تحسين الفعالية بالمقارنة مع الكلفة والموثوقية، حيث يعتبر النظام المتعدد المعالجات رائدا بين انظمة المعلوماتية حيث ينافس كل من نظام المعالج الوحيد والانظمة الموزعة على حد سواء. [2]

**4.1 حواسيب المعالجة المتوازية**

يحتوي هذا النوع من الحواسيب على مجموعة من المعالجات المستقلة والتي يحتوي كل منها على ذاكرة مستقلة او تشترك معا بذاكرة رئيسية، كما وتكون هذه المعالجات متزامنة، في هذا النوع يقوم الحاسوب بتقسيم المسائل الكلية الى مجموعة من المسائل الجزئية واعطاء كل مسألة من المسائل الى الى معالج حيث يتم حل تلك المسائل بشكل متزامن ومن ثم تجمع النتائج لاعطاء حل للمسألة الاصلية، وتقسم الاجهزة المتوازية الى قسمين رئيسيين حسب طريقة الاتصال بين وحدات المعالجة هي.[3]

**حواسيب الذاكرة المشتركة**

يعرف هذا النوع ايضا بحواسيب الوصول العشوائي المتوازي، حيث يحتوي هذا النوع من الحواسيب على ذاكرة مشتركة يتم تقاسمها من قبل مجموعة المعالجات الموجودة، وبنفس الطريقة وتسمى تلك التقسيمات بالـ blocks اي عندما تريد المعالجات الاتصال فيما بينها فان ذلك يتم من خلال الذاكرة المشتركة. حيث يسمح هنا لجميع المعالجات بالوصول الى الى الذاكرة المشتركة بنفس الوقت من اجل القراءة منها او الكتابة فيها، ويقسم هذا النوع من الحواسيب الى اربعة فئات رئيسية وفقا لقدرة اثنين او اكثر من المعالجات على الوصول الى موقع الذاكرة نفسه وبنفس الوقت.

Exclusive Read, Exclusive Write (EREW).

Concurrent Read, Exclusive Write (CREW).

Exclusive Read, Concurrent Write (ERCW).

Concurrent Read, Concurrent Write (CRCW).

**حواسيب شبكة الاتصالات**

هو عبارة عن نموذج مرن اكثر قوة من نموذج الذاكرة المشتركة، يكون فيه كل زوج من المعالجات متصل بخط ثنائي الاتجاه، حيث من الممكن لعدة ازواج ان تتصل بأن واحد ولكن بشرط ان لا يحاول اكثر من معالج ارسال البيانات الى معالج اخر او تلقي البيانات من معالج اخر.

**5.1 تعريف الخوارزميات المتوازية**

هي عبارة عن مجموعة من العمليات التي تنجز كل منها خوارزمية تتابعية خاصة بها وتتبادل الاتصالات فيما بينها عبر شبكة اتصالات محددة، وبما ان اتصال تلك المعالجات يتم بواسطة الذاكرة المشتركة القابلة للتقسيم لذلك فان اتصال كل معالج مع الاخر يتم بدون كلف زمنية.[4]

**6.1 تحليل الخوارزميات**

لقد ازدادت سرعة الحواسيب كثيرا في الاربعين سنة السابقة، فقد كان يعتقد بان فعالية الخوارزمية ليست ذات اهمية كبيرة، ولكن الحقيقة التي ظهرت فيما بعد اثبتت ان الفعالية امر مهم جدا، مما يدعونا الى التعمق في تحليل الخوارزميا ت المتوازية لمعرفة فعاليتها، حيث نقصد بالفعالية هنا هو درجة جودة الخوارزمية والتي تتحدد من خلال زمن تنفيذها وكلفة تنفيذها بالاضافة الى عدد المعالجات التي تحتاجها، في ادناه سوف نناقش تلك المعايير الثلاثة.[5]

**زمن التنفيذ**

ان سرعة الحسابات هو السبب الرئيسي الذي جعلنا نهتم ببناء الحواسيب المتوازية، لذلك يعتبر هذا المعيار هو المقياس الاهم في تقييم الخوارزمية المتوازية، ومن الممكن تعريفه بانه الزمن الذي تحتاجه الخوارزمية اثناء حلها على حاسوب متوازي، بمعنى اخر هو الزمن المستهلك من قبل الخوارزمية منذ اللحظة الاولى التي تبدأ فيها الى اللحظة التي تنتهي فيها.

**عدد المعالجات**

عند حل مسألة ما يجب حساب عدد المعالجات التي نحتاجها في حل تلك المسألة من اجل حساب كلفة الشراء والصيانة، فكلما زاد عدد المعالجات كلما ازدات الكلفة المدفوعة من اجل بناء حواسيب متوازية للحصول على ضمان درجات عالية من الوثوقية، على سبيل المثال من اجل حل مسألة ذات حجم (n) فأن عدد المعالجات التي نحتاجها هنا لبناء الخوارزمية المتوازية هو دالة بدلالة (n) اي P(n).

**كلفة التنفيذ**

من الممكن حساب كلفة بناء الخوارزمية المتوازية من خلال القانون التالي:

الكلفة = زمن التنفيذ \* عدد المعالجات المستخدمة

فمن اجل حل مسألة حجمها (n) فان كلفة بناء خوارزمية التوازي هي دالة بدلالة (n) اي C(n) وتحسب من المعادلة التالية:

C(n) = P(n) \* t(n)

حيث ان:

C(n): تمثل كلفة بناء الخوارزمية المتوازية.

P(n): تمثل عدد المعالجات المطلوبة.

T(n):تمثل الوقت المطلوب للمعالجة. **7.1 كفائة الخوارزمية**

تعرف كفائة الخوارزمية بانها ناتج ضرب عدد المعالجات التي تستخدم لانجاز عملية معينة والزمن التي تحتاجه لتنفيذ تلك العملية، حيث ان عملية تنفيذ الجمل الكبيرة من المعادلات الخطية على الكمبيوتر من الممكن ان يكلف الكثير من الجهد والوقت ومثال على ذلك انجاز عمليات الضرب والقسمة على كل من الاعداد الصحيحة والاعداد الحقيقية، لذا وجدت العديد من البحوث والخوارزميات لتقليل ذلك الجهد والوقت ومثال على تلك الخوارزميات هي خوارزمية غاوس والتي تقوم بحل جمل المعادلات الخطية ومن ثم تقوم بحساب الكلفة والجهد المبذول في حل تلك المعادلات والتي سنقوم بمناقشتها في بحوث لاحقة ان شاء الله.[6]

**2- المراجعة التاريخية**

**"كندة زين العابدين"، "خوارزميات المعالجة المتوازية وبرمجتها"، دراسة لنيل شهادة الماجستير / جامعة دمشق، سبتمبر 2006.**

ان تطور علم المعلوماتية منذ بدايته في منتصف القرن العشرين حتى بداية هذا القرن بشكل متسارع وكبير تشعبت فروعه وتعددت مجالاته واختصاصاته، واصبح دخول هذا العلم ضروريا في كل مجالات الحياة وكافة فروع العلوم الاخرى، حيث اقتضى ذلك تصميم خوارزميات ملائمة لحل المسائل والمشكلات المطروحة الى تطور كبير في علم الخوارزميات، واقتضى ذلك وضع خوارزميات تنفذ تتابعيا حيث تكون تعليمات تلك البرامج مرتبه ترتيبا دقيقا ويتم تنفيذها الواحدة تلو الاخرى، الا انه مع زيادة حجم المسائل الواجب حلها وتشعب المشكلات المطروحة وصعوبتها، اصبح من الصعب على تلك الخوارزميات الوصول الى حل دقيق لتلك المسائل والمشكلات اما بسبب تجاوز قدرات الذاكرة اللازمة لتخزين المعلومات او بسبب المدة الزمنية الكبيرة التي تحتاجها للوصول للحل الامثل.

من هنا ولدت مسألة التوازي في البرمجة الا انه تبين استحالة تنفيذها على الاجهزة التقليدية التي تعتمد على تقنية الكترونية ثابتة، الامر الذي استدعى التدخل في بنية الجهاز نفسه والبحث عن بنية جديدة وقد تم ذلك في مطلع الثمانينات من القرن العشرين تصميم اول حاسوب متعدد المعالجات دعي بـ CRAY1 والذي احتوى على معالجين يعملان في وقت واحد، مما استدعى ذلك تطوير خوارزميات دعيت بالمتوازية لتتلائم مع الواقع الجديد الذي فرضته تلك الحواسيب.[1]

**" محمد عبد الله، بسام عبد الرحمن، نواف فيصل، عمر صالح "، " الحاسبات المتوازية** **والخوارزميات المتوازية"، العربية السعودية 2005.**

لعدة سنوات خلت كانت الحاسبات المتوازية لا توجد الا في معامل الابحاث، اما اليوم فهذه الحاسبات متوفرة وعلى نطاق واسع في المجالات التجارية، حيث ان مجال المعالجة المتوازية قد نضج بما فيه الكفاية واصبح يدرس في المراحل الجامعية الاولى، كما ان تقنية التوازي تغطي طيفا واسعا من المكونات المادية بداية من تصميم ابسط تلك المكونات وهو الـ Adder وحتى تحليل النماذج النظرية للحسا المتوازي.

ان السبب الرئيسي لاستخدام التوازي في تصميم البرمجيات والمكونات المادية هو من اجل الحصول على الاداء الاعلى والسرعة العالية حيث ان كل انواع الـ Supper Computer اليوم تستخدم التوازي على نطاق واسع لزيادة سرعة الاداء، ومن امثلتها الحاسب الياباني الذي يدعى "محاكي الارض" والذي تم صناعته في عام 2003 حيث استخدم اكثر من خمسة الاف معالج تعمل بالتوازي.[3]

**"عبد الرحمن احمد محمد"، "Automatic Parallel Processing Environment"، جامعة الجوف.**

ان المعالجة المتوازية هي طريقة لتقسيم العمل إلى أجزاء غير مرتبطة مع بعضها البعض بحيث يمكن إنجاز كل جزء في معالج أو نواة، ويتم تنفيذ الأجزاء بالتوازي في وقت واحد، وقد بدأت فكرة التوازي منذ الستينيات من القرن الماضي حيث كانت تستخدم الحاسبات الكبيرة والتي تسمح لعدة طرفيات من التشارك في حاسب (Mainframes) ولكنها،(Time Sharing) واحد عن طريق التقسيم الزمني اشتهرت مع ظهور الحاسبات الشخصية ورخص سعرها وتوفرها في كل مكان.

ظهرت أول بيئة معالجة متوازية بصورة فعلية في بداية التسعينيات Parallel Virtual حيث ظهرت الآلة الإفتراضية المتوازية، ثم تلتها مكتبة واجهة تبادل الرسائل Machine (PVM) التي تعمل مع Message Passing Interface (MPI) لغات برمجة مختلفة مثل سي وجافا، وتوفرِ طرق تواصل بين الحاسبات عن طريق إرسال الرسائل.

بدأ تطبيق المعالجة المتوازية في المعامل المغلقة المرتبطة بشبكة محلية حيث تتوفر الموثوقية بين كل الأطراف، ثم تطورت الفكرة وتوسعت لتستفيد من حاسبات بعيدة في شبكة واسعة فيما يسمى الحوسبة الموزعة وظهر مع هذا التوسع الكثير من المفاهيم،(Mobile Computing) والمصطلحات مثل الحوسبة المتنقلة الحوسبة السحابية، (Volunteer Computing) حوسبة التطوع Grid الحوسبة الشبكية،(Cloud Computing)، (Utility Computing) والحوسبة الخدمية Computing وجميعها مبني على فكرة الحوسبة الموزعة ولكنها تختلف في الأسلوب وطرق الربط والتغطية الجغرافية. [2]

**"Guy E. Blelloch and Bruce M. Maggs"،"تحليل وتصميم الخوارزميات المتوازية"، Carnegie Mellon University".**

الخوارزميات هي عبارة عن مجموعة من الخطوات المتسلسلة التي تحدد تسلسل تنفيذ العمل وان كل خطوة من تلك الخطوات ستتضمن تنفيذ عملية احادية، حيث ان تلك الخوارزميات تكون ملائمة لتنظيم وتسهيل عمل حاسبات اليوم التي تقوم بتنفيذ الاعمال بشكل متسلسل، والذي يؤدي بدوره الى زيادة سرعة تنفيذ العمليات التي يقوم بها الكمبيوتر بالاضافة الى امكانية التوسع في الاجراءات والاعمال التي يقوم بها، ونظرا لدخول الكمبيوتر الى مجالات عديدة في الحياة وتوسع الاعمال التي يقوم بها كل ذلك اوجب ان تقوم تلك الحاسبات بتنفيذ اعمالها بسرعة وكفاءة عالية وباقل كلفة ممكنة، حيث ظهر ما يدعى بالحاسبات المتوازية والتي تعرف على انها تلك التي تؤدي عدة وضائف في نفس الوقت لحل كل الاوامر والمشاكل المعطاة بكفائة وفعالية بزمن اقل وبكلفة اقل.

ان عملية تصميم حواسيب متوازية اوجب بناء خوارزميات تنظم عمل تلك الحواسيب وتساعدها في تنفيذ اعمالها، حيث سميت تلك الخوارزميات بالخوارزميات المتوازية.[5]

**"Bernd Mohr "، "مقدمة الى الحواسيب المتوازية"، "Computational Nan** **science: Do It Yourself, NIC Series, Vol. 31, pp 491-505" (2006).**

ان المعالجة المتوازية هو جزء لا يتجزأ من الحياة اليومية، حيث ان هذا المفهوم نستخدمه في حياتنا اليومية من دون ان ندركه، فعندما نواجه مشكلة صعبة ونشرك الاخرين بحلها مما يؤدي الى سهولة حل تلك المشكلة بالاضافة الى سرعة حلها. حيث من الممكن وصف تلك الحالة بالمعالجة المتوازية، ومن الجدير بالذكر ان الاجهزة المستخدمة والمبتكرة من قبل اليونانيين قبل 2000 عام قد استغلت هذا المبدأ،
وفي القرن التاسع عشر الميلادي استخدم العالم Babbage مبدأ المعالجة المتوازية من اجل تحسين عمل المحرك الذي قام بتصميمه، وتجدر الاشارة بالذكر ايضا ان نظام التصميم الرقمي للكمبيوتر المعروف بـ ENIAC1 قد عرف حواسيب المعالجة المتوازية بانها نظام لامركزي عالي الاداء يتكون على الاقل من (25) حاسبة مستقلة تشترك جميعها من اجل حل مسألة معينة، حيث يتم تقسيم حل تلك المسألة بين تلك الحواسيب وفي النهاية تجمع النتائج لتكون الحل النهائي لتلك المسألة، حيث ظهر بالاعتماد على هذا المبدأ ما يعرف بحواسيب المعالجة المتوازية، ولاجل عمل تلك الحواسيب واستثمارها الاستثمار الامثل فقد تم تنظيم عملها باستخدام طرق وتقنيات حديثة تعرف بالخوارزميات المتوازية.[7]

**3- بناء الخوارزميات المتوازية**

**1.3 مفاهيم اساسية لبناء خوارزميات متوازية بكلف مثالية**

لو افترضنا ان زمن تنفيذ خوارزمية تسلسلية لانجاز مهمة حجمها (n) هو O(n) وزمن تنفيذ نفس المهمة على خوارزمية متوازية هو O(log n) حيث ان (n) تمثل عدد المعالجات المستخدمة في انجاز المهمة حيث يمكننا حساب الكلفة من خلال القانون التالي:

Cost =n\*O(Log n)=O(n\*Log n).

علما ان الكلفة التي ستنتج في القانون اعلاه هي ليست كلفة مثالية، وللحصول على خوارزمية تنفذ المهمة بكلف مثالية نحتاج احد امرين اما تخفيض زمن التنفيذ O(1) او انقاص عدد المعالجات الى n / Log n حيث ان ناتج ضربهما هو:

O (n) = n / Log n \* O(Log n)

ليكن كل معالج ياخذ Log(n) عنصر بيانات، فأذا كانت العملية التسلسلية خطية في عدد عناصر البيانات عندها كل من n / Log(n) هو معالج سينجز عملية تسلسية على التوازي على كل عناصر بياناتها Log(n) في زمن O(Log n).

اضافة الى ان المعالجات n / Log n ستنجز الخوارزمية السابقة (غير المثالية) على النتائج n / Log n من الخطوة التسلسلية في.

Log (n / Log n) = Log n – Log (Log n) = O (Log n)

ومنه فان خطوة كل كلفة هي:

O (Log n)

والكلفة هي:

(n / Log n) \* O (Log n) = O (n)

وهو المطلوب.[8]

**2.3 تصميم خوارزميات المعالجة المتوازية**

ان عملية تصميم خوارزمية متوازية هي عملية حركية حيوية، حيث ان الشكل الطبيعي لتطوير الخوارزميات المتوازية يتم من خلال البدء من الخوارزمية التسلسلية ومن ثم تحديد الخطوات المستقلة للحصول على خوارزمية متوازية، وان هذا التحول من التسلسل الى التوازي ينتج عنه في اغلب الاحيان خوارزميات متوازية ذات كلف منخفضة ومقبولة، ولغرض الحصول على خوارزميات متوازية فعالة هنالك عدد من التقنيات التي تستخدم لهذا الغرض من اهمها:

ملاحظة هامة: سنرمز للزمن هنا بالرمز (T) والمكونات المادية بالرمز (H) وعدد المعالجات بالرمز (P) اما الزمن التي تحتاجه المعالجات في تنفيذ عملها فهو (PT).

**التقسيم والتخصيص**

تعتمد هذه التقنية على تجزئة المسألة الى مسائل جزئية مستقلة من نفس النوع يتم حلها على التوازي وتجمع حلولها للحصول على حل للمسألة الاصلية.

Let (r) is integer number divide (n).

And let Hi (n/r) and Ti (n/r) such that (i=1, 2, 3, 4,…….., r).

Is a cost of solute (r) to partial problem?

Let Tc(n)، Hc(n) is additional parallel cost to solute the original problem after to solute a partial problem, such that:

T(n) <= Max (T1(n/r), T2(n/r),………., Tr(n/r) + Tc(n))……..(1).

H(n) <= Max (H1(n/r) + H2(n/r) + …….. + Hr (n/r), Tc(n)) ………. (2).

Such that the relation no. (1) Will conclude from the following facts:

Problem individual.

Need to additional work when we solute the partial problems only.

And the relation no. (2) Will conclude from the following facts:

ان كل مسألة جزئية تم حلها على التوازي مع الاخريات تحتاج الى مكونات مادية.

ان الكمية الناتجة من المكونات الصلبة تمثل العدد الاكبر من المصادر المطلوبة لحل جميع المسائل الجزئية والمصادر المطلوبة للوصول الى الحل الاخير للمسألة الاصلية.

وعلى افتراض ان جميع المسائل الجزئية متساوية من حيث كلفة التوازي تصبح العلاقتين (1) و (2) كما يلي:

T(n) <= T(n/r) + Tc (n) …. (3).

H(n) <= Max (r \* H(n/r)، Hc(n)) ……….. (4).

حيث ان T(n/r) و H(n/r) هي كلفة المسألة الجزئية من الترتيب (n/r)، وتعتبر هذه العلاقات من (1.... 4) جدا مهمة من اجل تحليل الخوارزميات المتوازية التي نحصل عليها من طريقة التصميم والتخصيص.

**طريقة استخدام المتجه**

تبدأ هذه التقنية من تحليل المسألة وانتاج الخوارزميات التي تعمل على بناء المعطيات التي تتناسب مع حسابات المتجه.

**تكرار المتجهات**

قام بوضع اسس هذه التقنية العالم Traub واخرون من اجل حل الجمل الخطية المثلثية، حيث ستقوم باستخدام الخوارزميات المتسلسلة المباشرة ونبني منها الطريقة التكرارية المتوازية، فاذا قمنا بأخذ الطريقة التكرارية لحساب عناصر مصفوفة مثلثية بالشكل التالي.

X1 = a1.

Xi = ai – (bi \* ci / xi – 1). Such that 2 <= i <= n.

عندئذ من الممكن ان نحصل على التكرارات التالية وفق طريقة Gaues.

Xi (0) = a1.

Xi (j) = ai – (bi \* ci / xi – 1 (j-1)).

حيث ان (j) هو رقم التكرار.

كما ويمكن ان تكون هذه التكرارات شعاعية، اي انه من الممكن ان تفترض تعديلا للشعاع (x(j)) حيث ان (j >=0)

نجد انه في الخطوة الاولى (j=0) كل عنصر في الشعاع (X(0)) تحتوي القيمة (a1).

اما في الخطوة الثانية فيحتوي العنصر رقم (i) لـ (X1)) على القيمة التالية.

ai – (bi \* ci / xi-1(0)) = ai – (bi \* ci / a1)، i = 1,2,3,…..,n.

عندئذ من الممكن ان نعدل كل العناصر في (x(1)) بوقت واحد معا وهذا يطبق المبدأ نفسه على التكرارات التسلسلية الجزئية.

**المضاعفة التعاودية**

هي عملية تحويل البيان الحسابي من بيان بدرجة تعقيد خطية موضحة في الشكل رقم (1) الى بيان بدرجة تعقيد لوغارتمية موضحة في الشكل رقم (2)، وهذا يمكننا من الحصول على بيان تكون فيه درجة التعقيد تابع اسي (لوغارتمي) من بيان درجة التعقيد فيه تابع خطي.

شكل رقم (1) بيان بدرجة تعقيد خطية

شكل رقم (2) بيان بدرجة تعقيد لوغاريتمية

ففي الشكل اعلاه من اجل حجم n = 8 نلاحظ ان عمق البيان الحسابي في الشكل رقم (1) هو (7) اما في الشكل رقم (2) فهو (3).[9]

**3.3 تحويل الخوارزميات الخطية الى متوازية**

لنأخذ حساب الـ Xn حيث ان n = 2k من اجل عدد صحيح (k) حيث ستكون الخوارزمية له بالشكل التالي.

Begin

S  X;

For I 1 To (n-1) Do

S  S \* X;

End.

نلاحظ ان البيان الحسابي الناتج عن هذه الخوارزمية له درجة تعقيد خطية (عمق خطي) وهو موضح في الشكل رقم (3).



شكل رقم (3) يوضح العمق الخطي للخوارزمية اعلاه

اما اذا اردنا الحساب باستخدام درجة التعقيد اللوغارتمية فستكون حسب الخوارزمية التالية.

Begin

S  X;

For I 1 To K Do

S  S \* S;

End.

عندئذ سنحصل على البيان الموضح في الشكل التالي رقم (4). وهو بدرجة تعقيد لوغارتمية.

حيث يوضح الشكل (A) بيان حسابي من اجل حساب تسلسلي لـ (X8) بدرجة تعقيد خطية، ويوضح الشكل (B) بيان حسابي من اجل حساب تسلسلي لـ (X8) بدرجة تعقيد لوغارتمية.[10]



شكل رقم (5) يوضح العمق اللوغاريتمي للخوارزمية اعلاه

**4.3 امثلة على الخوارزميات المتوازية**

**خوارزمية التحقق من التوزيع (خوارزمية النشر)**

لقد ذكرنا في الجزء الاول من بحثنا (المقدمة) انواع حواسيب المعالجة المتوازية وقسمناها الى قسمين هما حواسيب الذاكرة المشتركة وحواسيب شبكة الاتصالات، وفي نفس الوقت قمنا بتقسيم حواسيب الذاكرة المشتركة الى اربعة انواع وفقا لقدرة اثنين او اكثر من المعالجات على الوصول الى موقع الذاكرة نفسه في نفس الوقت. ومن تلك الانواع ما سمي بالـ Exclusive Read Exclusive Write (EREW) حيث لا يمكن في هذا النوع الوصول الى نفس موقع الذاكرة لمعالجين معا في نفس الوقت، ومع ذلك يكون من الضروري ان تقرأ عدة معالجات عنصر بيانات محدد محفوظ في موقع معين من الذاكرة المشتركة.

بمعنى اخر اننا قد نحتاج الى نشر عنصر بيانات ما على جميع المعالجات في الحاسب المتوازي من اجل استعماله من قبلها جميعا في نفس الوقت.

ومن الواضح ان المحاكاة الفعالة لهذه العملية لا يمكن انجازها في خطوة واحدة في نموذج EREW ونحتاج من اجل تلك المحاكاة الى وضع اجراء ينفذ هذه المهمة، حيث ان عملية بناء هذا الاجراء ستتم كما يلي.[1]

**بناء خوارزمية النشر**

ليكن لدينا ملف كبير جدا يتكون من (m) ادخال منفصل، ولنفترض ان الملف غير مرتب باي طريقة، فاذا اردنا البحث عن عنصر معين مثل (x) في هذا الملف على حاسوب من النموذج EREW مع (n) معالج، حيث ان (n < m) عندئذ نحتاج الى نشر هذا العنصر على جميع المعالجات ولنرمز لتلك المعالجات بالرمز P1,P2,…,Pn. حيث ستتم عملية النشر باستخدام الاجراء التالي.

المعالج (P1) يقرأ (x) ويبلغ (P2).

بنفس الوقت المعالج P1,P2 تبلغ P3,P4 على التوالي.

بنفس الوقت المعالج P1,P2,P3,P4 تبلغ P5,P6,P7,P8 على التوالي.

وهكذا تستمر العملية حتى تحصل جميع المعالجات على (x) حيث يتضاعف عدد المعالجات في كل مرحلة مما يعني ان هذه العملية تتطلب (log n) خطوة، لنفترض ان جميع المعالجات تحتاج عنصر بيانات معين في لحظة ما خلال تنفيذ الخوارزمية وليكن هذا العنصر محجوز في الموقع (D) من الذاكرة، حيث ستتم عملية محاكاة القرأة المتعددة هذه عن طريق عملية النشر والتي تعرف بالاجراء Broadcast حيث يفترض في هذا الاجراء وجود مصفوفة اسمها (A) طولها (N) في الذاكرة، حيث ستكون تلك المصفوفة فارغة في البداية ويستعملها الاجراء كمكان لنشر محتويات (D) الى المعالجات، ويرمز لها بالـ A(i).

**خطوات خوارزمية النشر**

الخطوة رقم (1)

اقراالقيمة الموجودة في (D).

خزنها في الذاكرة الخاصة بـ (P1).

اكتبها في A(1).

الخطوة رقم (2).

From(i=0) To (log N-1)

From (j=2i+1) To (2i+1)

المعالج (Pj).

اقرأ القيمة في A(j-2i).

خزنها في الذاكرة الخاصة بـ (Pj).

اكتبها في A(j).

حيث ستكون الخوارزمية كالتالي.

Void BROUDCAST (D, N, A)

Step1: Process P1

Read the values in (D).

Store it in its own memory.

Write it in A (1).

Step2:

For (i=0 ; I < = log (N-1) ; i++)

For (j=2i+1 ; j<=2i+1 ; j++)

Process Pj

Read the value in A(j-2i).

Store it in its own memory.

Write it in A (j).

سيوضح الشكل التالي (5) الاجراء Broadcast من اجل N=8، D=5.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | **5** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | **5** |

**5**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

**D**

**A**

الشكل رقم (5- أ) يوضح تنفيذ الخطوة رقم (1) في الخوارزمية

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **5** | **5** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | **5** | **5** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

**A**

الشكل رقم (5- ب) يوضح تنفيذ الخطوة رقم (2) عندما (i=0)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **5** | **5** | **5** | **5** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **5** | **5** | **5** | **5** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

**A**

الشكل رقم (5- ج) يوضح تنفيذ الخطوة رقم (2) عندما (i=1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P8 | P7 | P6 | P5 | P4 | P3 | P2 | P1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

**A**

الشكل رقم (5- د) يوضح تنفيذ الخطوة رقم (2) عندما (i=2)

عندما ينتهي الاجراء تكون جميع المعالجات قد خزنت القيمة (D) في ذاكرتها المحلية الخاصة من اجل استخدام اخر، طالما ان المعالجات المعتمدة قد قرأت (D) مرتين في كل تكرار فان الاجراء ينتهي في زمن O Log(n).

ان الذاكرة المطلوبة من الاجراء Broadcast هي مصفوفة طولها (N) وبكلام ادق مصفوفة من نصف ذلك الطول اذ انه في اخر تكرار للاجراء تكون جميع المعالجات قد استقبلت القيمة في (D) ولا تحتاج ان تكتبها مرة اخرى في (A). وهي موضحة في الشكل اعلاه (5-د).

يمكن للاجراء Broadcast ان يتكيف بسهولة لمنع الكتابة الاخيرة ومن اجل ذلك يستخدم مصفوفة (A) بطول (N/2).

 **4. النتائج**

لقد رأينا انه من المفيد ان نختتم بحثنا هذا ببعض النتائج التي توصلنا اليها، ومن تلك النتائج:

* ان عملية دراسة الخوارزميات المتوازية، هذا الفرع الجديد من فروع المعلوماتية تعتبر مسألة صعبة في وقتنا الحاضر وذلك لعدم توفر اجهزة ملائمة تسمح بترجمة الخوارزميات المصممة الى برامج عملية نتوصل من خلال تطبيقها على نتائج واقعية تخدم حياتنا العملية.
* ان تطبيق مبدأ المعالجة المتوازية على اجهزة متعدةة المعالجات يساعدنا في زيادة حجم المسائل الممكن حلها، كما تساعدنا في الحصول على معلومات اكثر وحلول افضل من خلال تقليل كلفة حل المسائل.
* ضرورة تخصيص معالجات بحجم كبير لاستثمار الحواسيب المتوازية بشكل افضل.
* ان استخدام تقنية الموازاة يؤدي الى تقليل زمن الوصول الى الحل الامثل. اذ انها تمكن من الوصول بشكل مبكر الى الحل وبالتالي تقلل من زمن التنفيذ.

**5. التوصيات**

ان الخوارزميات المتوازية هي فرع جديد وهام من فروع المعلوماتية، كما انها تفتح الابواب على مصراعيها امام الباحثين والمهتمين في هذا المجال لذلك ندعوا جميع الباحثين لمتابعة هذا البحث سواء في المجال النظري او المجال العملي، لاننا نعتقد ان هذا النوع من فروع المعلوماتية سيبقى لوقت طويل موضوع الساعة ومحور اهتمام كافة المختصين والباحثين.

**المصادر**

**المصادر العربية**

كندة زين العابدين، خوارزميات المعالجة المتوازية وبرمجتها، دراسة لنيل شهادة الماجستير / جامعة دمشق، سبتمبر 2006.

عبد الرحمن احمد محمد، Automatic Parallel Processing Environment، جامعة الجوف.

محمد عبد الله، بسام عبد الرحمن، نواف فيصل، عمر صالح، الحاسبات المتوازية والخوارزميات المتوازية، العربية السعودية 2005.

شيرين فؤاد حبيب، بناء محاكي للمعالجات المتوازية، جامعة تشرين 2015

**English Resources.**

Guy E. Blelloch and Bruce M. Maggs, Analyze and design of the parallel algorithm, Carnegie Mellon University.

Ananth Grama, Anshul Gupta, Principles of Parallel Algorithm Design, To ccompany the text "Introduction to Parallel Computing", Addison Wesley, 2003.

Bernd Mohr, Introduction to the parallel computers, Computational Nan science: Do It Yourself, NIC Series, Vol. 31, pp 491-505 (2006).

Denis Caromel, Arnaud Contes, Parallel Processing, Pro Active Journal 2009.

John L. Hennessy, David A. Patterson, Computer Architecture A quantitative Approach, Fourth Edition, Morgan Kaufmann.

Chris Davis, Sonja Keserovic, Parallel Processing, International Journal of engineering and advance technology.

Keshab K. Parhi, Pipelining and Parallel Processing, International Journal of engineering and advance technology.

Parallel Computing on Wikipedia. [http://en.wikipedia.org/wiki/Paral­lel\_computing](http://en.wikipedia.org/wiki/Parallel_computing).

Paul Austin, Kevin Murray, Andy Wellings. The Design of an Operating System for a Scalable Parallel Computing Engine. <https://www.cs.ubc.ca/cs-research>

Jeffrey Richter, Concurrency and Coordination Runtime, https://msdn.microsoft .com /magazine/msdn-magazine-issues