

Enhancement of Power Consumption of Tasks Scheduling Algorithms on a Multi-Core Platform in Real Time Systems

Dr. Mohammed Hijazieh*
Youssef Ntefeh**

(Received 3 / 10 / 2021. Accepted 15 / 2 / 2021)

□ ABSTRACT □

Real-time systems are the basic core of most research fields in both academic and applied directions, and the reason for this is due to the need to provide a high quality of service for these fields as a result of the rapid and increasing progress of information and communication technology, in addition to the rapid development in the design of processors and electronic controllers, which have high processing speed and the ability to Communicate with a very wide range of other peripheral units.

The basic core in the success of any system that works in real time is its possession of an accurate and tight scheduling algorithm capable of dealing with various types of tasks, such as periodic, sporadic and aperiodic, where the effectiveness of the algorithm lies in ensuring the execution of these tasks in a logical manner without exceeding the associated time constraints.

Recently, researchers have tended to study the possibility of the real-time system working in power-saving conditions due to the large spread of portable devices and the dependence of these devices in their work on real-time systems, and this research aims to provide development and improvement on the work of scheduling algorithms in terms of energy consumption during the operation of the system.

In this research, an integrated software system for the LLREF scheduling algorithm was designed and built using the MATLAB integrated development environment, based on the mathematical model of each of the algorithm, tasks, multi-core platform and energy consumption equations.

Keywords: Scheduling, periodic tasks, sporadic tasks, real time systems, energy consumption.

* Professor, Department of computer and automatic control Engineering, Faculty of Mechanical and electrical Engineering, Tishreen University, Latakia, Syria. mohammed.hejazieh2016@gmail.com

** PhD Student, Department of computer and automatic control Engineering, Faculty of Mechanical and electrical Engineering, Tishreen University, Latakia, Syria. Email: youssef.ntefeh@gmail.com

تحسين استهلاك الطاقة في خوارزميات جدولة المهام على منصة عمل متعددة النوى في نظم الزمن الحقيقي

د. محمد حجازية*

يوسف نتيفة**

(تاريخ الإيداع 3 / 10 / 2021. قَبْلَ للنشر في 15 / 2 / 2021)

□ ملخص □

تعد نظم الزمن الحقيقي النواة الأساسية لمعظم المجالات البحثية بشقيها الأكاديمي والتطبيقي، ويعود السبب في ذلك إلى الحاجة إلى تأمين جودة خدمة عالية لهذه المجالات نتيجة التقدم السريع والمتزايد لتكنولوجيا المعلومات والاتصالات، بالإضافة إلى التطور المتسارع في تصميم المعالجات والمتحكمات الالكترونية والتي تتمتع بسرعة معالجة كبيرة وبقدرة على التواصل مع شريحة واسعة جداً من الوحدات المحيطية الأخرى.

إن النواة الأساسية في نجاح عمل أي نظام يعمل بالزمن الحقيقي هي امتلاكه لخوارزمية جدولة دقيقة ومحكمة وقادرة على التعامل مع مختلف أنواع المهام سواءً الدورية منها أو العشوائية أو غير الدورية، حيث تكمن فعالية الخوارزمية في تأمين تنفيذ هذه المهام بالشكل المنطقي السليم دون تجاوز القيد الزمني المرتبط بها.

اتجه الباحثون في الفترة الأخيرة إلى دراسة إمكانية عمل نظام الزمن الحقيقي في ظروف توفير الطاقة نظراً لانتشار الأجهزة المحمولة بشكل كبير واعتماد هذه الأجهزة في عملها على نظم الزمن الحقيقي، وهذا البحث يهدف إلى تقديم تطوير وتحسين على عمل خوارزميات الجدولة من حيث استهلاكها للطاقة أثناء تشغيل النظام.

تم في هذا البحث تصميم وبناء نظام برمجي متكامل لخوارزمية الجدولة LLREF باستخدام بيئة التطوير المتكاملة MATLAB، وذلك بالاعتماد على النموذج الرياضي لكل من الخوارزمية والمهام ومنصة العمل متعددة النوى ومعادلات استهلاك الطاقة، وتوصل البحث إلى تحقيق آلية جديدة لعمل الخوارزمية ضمن إطار توفير استهلاك الطاقة.

الكلمات المفتاحية: الجدولة، المهام الدورية، المهام العشوائية، نظم الزمن الحقيقي، استهلاك الطاقة.

* أستاذ - قسم هندسة الحاسبات والتحكم الآلي - كلية الهندسة الميكانيكية والكهربائية - جامعة تشرين - اللاذقية - سورية.

mohammed.hejazieh2016@gmail.com

** طالب دكتوراه - قسم هندسة الحاسبات والتحكم الآلي - كلية الهندسة الميكانيكية والكهربائية - جامعة تشرين - اللاذقية - سورية.

youssef.ntefeh@gmail.com

مقدمة:

تعد جدولة المهام من المواضيع الأساسية في نظم الزمن الحقيقي، ولها عدة تصنيفات تبعاً لطبيعة منصة العمل سواءً كانت بمعالج وحيد أو متعددة المعالجات أو متعددة النوى، وطبيعة المهام التي تجدولها، وأسلوب ترحيل المهام، وهل طريقة عملها ساكنة أم ديناميكية.

نقول عن خوارزمية الجدولة أنها أمثلية Optimized إذا كانت قادرة على جدولة المهام دون تجاوز القيد الزمني Deadline المرتبط بهذه المهام، فقد تكون الخوارزمية أمثلية عند العمل على منصة عمل بمعالج وحيد وغير أمثلية عند العمل على منصة متعددة المعالجات أو متعددة النوى، وبالمثل فقد تكون أمثلية عند جدولة المهام الدورية، لكن عند ورود مهام عشوائية أو غير دورية تصبح غير أمثلية.

تم في هذا البحث تحسين عمل خوارزمية الجدولة Largest Local Remaining Execution First (LLREF) والتي تعتبر أمثلية عند العمل على معالج وحيد أو معالجات متعددة، وكذلك الأمر عند ورود أنواع مهام مختلفة دورية أو عشوائية أو غير دورية.

أهمية البحث وأهدافه:

يهدف هذا البحث إلى تطوير عمل خوارزمية الجدولة LLREF ضمن إطار توفير استهلاك الطاقة، بهدف تحسين أداء نظم الزمن الحقيقي وجعله يعمل لفترة زمنية أطول وبالتالي زيادة كفاءة وفعالية النظام بشكل عام.

طرائق البحث ومواده:

سوف نستعرض مقدمة نظرية موجزة عن نظام الزمن الحقيقي، وخوارزميات الجدولة والنموذج الرياضي المستخدم في هذا البحث.

1. نظام الزمن الحقيقي

نطلق مصطلح نظام الزمن الحقيقي على أي نظام تشغيل أو تحكم أو مراقبة بحيث تكون المهام المسؤولة عن تنفيذها مرتبطة بقيد زمني Deadline، يجب عليه تأمين تنفيذها بالشكل السليم ضمن هذا القيد، حيث يُعد تجاوز القيد الزمني بمثابة إخفاق لعمل النظام، وتُقسم نظم الزمن الحقيقي إلى قسمين رئيسيين هما:

- نظم الزمن الحقيقي القاسية Hard Real Time Systems

تُعتبر هذه النظم صارمة تجاه تجاوز المهام لقيدها الزمني، فأى تجاوز يشكل إخفاقاً لعمل النظام ككل، وكمثال على هذا النوع من النظم هو نظام يحتوي على مهمة قراءة حالة حساس الحريق في منشأة صناعية، عندها يشكل أي تأخر في قراءة حالة الحساس إلى انتشار الحريق وصعوبة السيطرة عليه مما يتسبب للمنشأة بأضرار بليغة.

- نظم الزمن الحقيقي المرنة Soft Real Time System

تُعتبر هذه النظم متساهلة مع تجاوز المهام للقيد الزمني المرتبط بها، ولا يشكل هذا التجاوز خطراً على عمل النظام، مثل نظام إرسال إشعارات الطقس للزبائن، فتجاوز القيد الزمني لهذه المهمة لن يؤدي إلى فشل عمل النظام ككل، إنما سيؤدي إلى تأخير في إيصال معلومة حالة الطقس للزبائن وهذا يترتب عليه كلفة أو جزاء Penalty تتناسب طردياً مع حجم التأخير الزمني بعد التجاوز. [3][12]

2. خوارزمية الجدولة LLREF

ينتمي الجدول الخاص بهذه الخوارزمية إلى النوع الحركي أي يتم استدعاؤه لترتيب أفضليات المهام بشكل حركي أثناء تشغيل النظام، وهو مقاد بالأحداث Event Driven بمعنى أنه يُستدعى لترتيب المهام عند ورود حدث معين. [4] [14]

وتُستند هذه الخوارزمية من حيث طبيعة إسناد الأفضليات إلى صنف الإسناد الحركي على مستوى المهام؛ وهو ما يقتضي إمكانية تنفيذ المهمة الحالية على أكثر من معالج موجود في المنصة دون الحاجة إلى التقيد بمعالج معين، والحركي على مستوى الأعمال؛ وهو ما يقتضي إمكانية أن يستأنف العمل التابع للمهمة الحالية تنفيذه إما على نفس المعالج الذي بدأ التنفيذ عليه قبل استبعاده بسبب الأفضليات أو على أي معالج آخر موجود في منصة العمل. [10]

وتُستند هذه الخوارزمية من حيث طبيعة ترحيل المهام إلى صنف الترحيل الكامل؛ وهو ما يقتضي إمكانية تنفيذ الأعمال المتتالية للمهمة الحالية على أي معالج موجود في المنصة. والأحداث في هذه الخوارزمية تنقسم إلى قسمين هما الأحداث الرئيسية، والأحداث الثانوية. أما الأحداث الرئيسية في هذه الخوارزمية تشمل حدثين هما:

➤ حدث إطلاق مهمة Task Release Event: وهو الحدث الناتج عن إطلاق مهمة معينة وانضمامها إلى رتل المهام الذي يُخدم من المعالجات الموجودة في المنصة.

➤ حدث إنهاء تنفيذ مهمة End of Task Execution Event: وهو الحدث الناتج عن إنهاء تنفيذ المهمة الحالية من قبل أحد معالجات المنصة، الأمر الذي يؤدي إلى خروجها من رتل المهام، وإتاحة الفرصة لمهمة جديدة لدخول الرتل. [8]

وتشمل الأحداث الثانوية Secondary Events حدثين هما:

➤ حدث نفاذ سماحية الوقت أو الحدث C: وهو الحدث الناتج عن عدم تبقي وقت إضافي للتنفيذ يمكن معه تأخير تنفيذ المهمة الحالية بحيث لا تخرق قيدها الزمني.

➤ حدث وجود حيز زمني إضافي للتنفيذ أو الحدث B: وهو الحدث الناتج عن إنهاء المهمة الحالية لكمية العمل التي يجب عليها إنجازها خلال مخطط (TL-Plane) Time and Local execution time Plane الأمر الذي يقتضي استدعاء الجدول ضمن المخطط بشكل استثنائي لجدولة المهام من جديد نتيجة لإمكانية إضافة مهمة جديدة إلى الرتل. [5][8]

1-2 نموذج خوارزمية LLREF

تعتمد هذه الخوارزمية أسلوب الجدولة العامة، حيث أن مستوى ترحيل المهام غير محدد بين مجموعة من المعالجات المتطابقة في السرعة والمكونات (SMP (Symmetric Multi-Processors) وسوف نفترض أن النظام سيحوي على مجموعة من المهام:

$$T = \{T_1, T_2, \dots, T_N\} \quad (1)$$

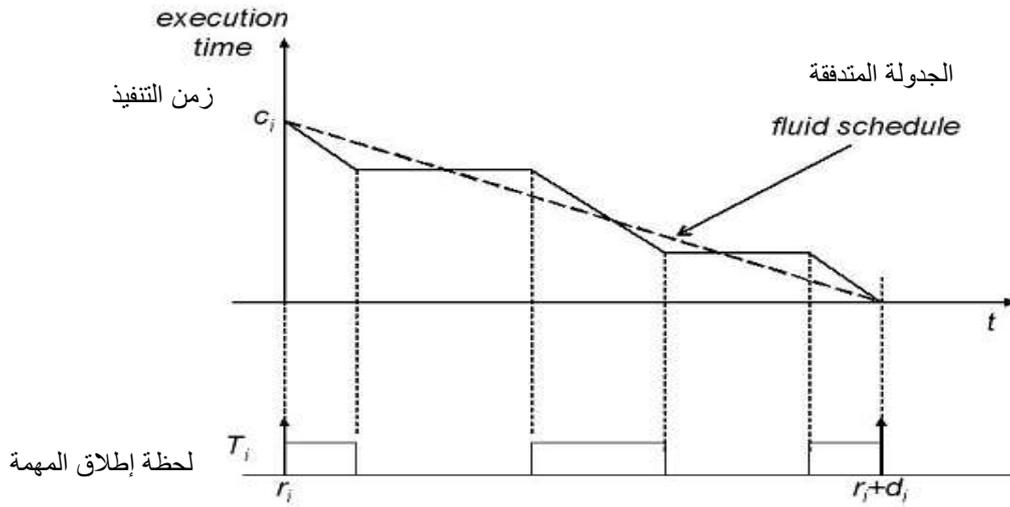
وهذه المهام كلها دورية حيث يتم إطلاقها عند اللحظة r_i ، كل مهمة T_i لها زمن تنفيذ c_i ، والقيد الزمني الخاص بها d_i والذي يساوي للدور p_i ، ونسبة استخدام كل مهمة للمعالج $u_i = \frac{c_i}{d_i}$ ، والتي هي أقل من الواحد. وسوف نفترض أيضاً بأن المهام مستقلة عن بعضها البعض أي أنها لا تتشارك بأي موارد ولا يوجد بينها اعتماديات، وبأنها ذات أفضليات مختلفة حيث يمكن في أية لحظة إزاحة المهمة الحالية من أجل تنفيذ مهمة ذات أولوية أعلى.

كما أن نظام العمل لا يعتمد على التوفير في استهلاك الموارد أي يمكن أن يكون لدينا معالجات شاغرة مع وجود مهام تنتظر التنفيذ، وسنعتبر أن الكلفة أو الأعباء الناتجة عن تبديل السياق وترحيل المهام بين المعالجات مهمة. [1][13]

2-2 مخطط الـ TL-Plane (Time and Local execution time Plane)

تتخذ كل مهمة من المهام بمعدل ثابت مع مرور الزمن في حال كان لدينا جدولة متدفقة، نجحت خوارزمية الـ Proportional-fair (PF) في الوصول إلى جدولة مثلى باستخدام هذه الطريقة حيث أن الجدولة المثلى تعني تحقيق كل القيود الزمنية للمهام الموجودة، ولكن لتجنب الفترات الزمنية التي تفرضها خوارزمية الـ PF سنلجأ إلى مفهوم جديد هو مخطط الـ TL حيث يتم تمثيل كل مهمة على اعتبارها كعلامة Token يتحرك باستمرار ضمن هذا المخطط مع مرور الزمن، ويتميز هذا المخطط بقدرته على مواكبة التنفيذ التدفقي للمهام من دون الحاجة إلى استخدام الفترات الزمنية. [9]

يوضح الشكل (1) الفكرة الأساسية من الجدولة المتدفقة والتي يعتمد عليها مخطط الـ TL حيث أن كل مهمة $T_i(r_i, c_i, d_i)$ تمثل ضمن شكل ثنائي الأبعاد، ويكون المحور الأفقي هو محور الزمن، والمحور الشاقولي هو الزمن المتبقي من تنفيذ المهمة.



الشكل (1) الجدولة المتدفقة [7]

3- النموذج الرياضي للمهام Mathematical Model of Tasks

يمكن صياغة النموذج الرياضي للمهام في هذا البحث على النحو التالي:

بفرض أنه لدينا رتل من المهام (Q) يحتوي على مجموعة من المهام الدورية (Periodic) (P) والعشوائية (Sporadic) (S) وغير الدورية (Aperiodic) (A) وذلك من أجل جدولتها على معالج متعدد النوى Multi-cores Processor فإن نموذج المهام الدورية Periodic Tasks Model يوصف كما يلي:

1. زمن ورود المهمة (t_p): ويوصف بأنه الزمن الذي يتم عنده رفع إشعار أو إعلام بورود حدث جديد

في نظام الزمن الحقيقي، حيث يشترط ورود المهام وفقاً لأدوار زمنية متساوية، أي:

$$\forall P \in Q \Rightarrow t_p(n+1) = t_p(n) + \tau \quad (2)$$

حيث: (τ) هو دور المهمة.

(n): عدد صحيح يمثل رقم الدور.

2. زمن تنفيذ المهمة الدورية (E_p) Execution Time: وهو زمن إشغال المعالج من قبل المهمة لكي تنفذ.

3. القيد الزمني للمهمة الدورية (D_p) Deadline: ويوصف بأنه الزمن الذي يتوجب على المهمة إنهاء

التنفيذ قبل بلوغه.

4. مع التنبؤ على أن المهام الدورية في هذا البحث ذات قيد زمني ضمني Implicit Deadline أي:

$$\forall P \in Q \Rightarrow \tau_p = D_p \quad (3)$$

ويكون نموذج المهام العشوائية Sporadic Task Model كما يلي:

1. زمن ورود المهمة العشوائية (t_s): وهو الزمن الذي يتم عنده رفع إشعار أو إعلام بورود حدث جديد

في نظام الزمن الحقيقي، ويشترط عدم ورود حدثين متتاليين لنفس المهمة في نفس اللحظة في هذا النوع من

المهام، أي:

$$\forall S \in Q \Rightarrow t(n)_s \neq t(n+1)_s \quad (4)$$

2. زمن تنفيذ المهمة العشوائية (E_s) Execution Time: وهو زمن إشغال المعالج من قبل المهمة لكي تنفذ.

3. القيد الزمني للمهمة العشوائية (D_s) Deadline: ويوصف بأنه الزمن الذي يتوجب على المهمة إنهاء

التنفيذ قبل بلوغه.

4. عدد مرات ورود المهمة العشوائية (N_s): وهو عدد يدل على عدد مرات إطلاق المهمة العشوائية

خلال مدة التنفيذ.

ويكون نموذج المهام غير الدورية Aperiodic Task Model كما يلي:

1. زمن ورود المهمة غير الدورية (t_a): وهو الزمن الذي يتم عنده رفع إشعار بورود حدث جديد في

نظام الزمن الحقيقي، ويمكن ورود حدثين متتاليين لنفس المهمة بفارق زمني أكبر أو يساوي الصفر؛ أي:

$$\forall A \in Q \Rightarrow t(n)_a \leq t(n+1)_a \quad (5)$$

2. زمن تنفيذ المهمة غير الدورية (E_a) Execution Time: وهو زمن إشغال المعالج من قبل المهمة لكي تنفذ.

3. القيد الزمني للمهمة غير الدورية (D_a) Deadline: ويوصف بأنه الزمن الذي يتوجب على المهمة

إنهاء التنفيذ قبل بلوغه.

4. عدد مرات ورود المهمة العشوائية (N_A): وهو عدد يدل على عدد مرات إطلاق المهمة العشوائية خلال مدة التنفيذ. [11] [12]

4- نموذج المعالج متعدد النوى Multicores Processor Model

نموذج المعالج المستخدم في البحث هو معالج متعدد النوى يحتوي على نواتين أو أربع نوى أو ثماني نوى وذلك وفق السيناريو المدروس، بحيث تكون هذه النوى متطابقة Identical، ويكون لكل نواة ذاكرة كاش من المستوى الأول L1 Cache Memory خاصة بها، وتشارك النوى بالمستوى الثاني من ذاكرة الكاش L2 Cache Memory.

5- النموذج الرياضي لاستهلاك الطاقة Power Consumption Model

تعرف الطاقة الكلية المستهلكة في المعالج على أنها عبارة عن مجموع كل من الطاقة الديناميكية والطاقة الساتاتيكية.

$$P_{dissTotal} = P_{Dynamic} + P_{Static} \quad (6)$$

$$P_{Dynamic} = C \cdot V^2 \cdot F \quad (7)$$

V: الجهد المطبق على المعالج.

F: تردد عمل المعالج.

C: ثابت التناسب.

$$P_{Static} = V \cdot I_{Le} \quad (8)$$

V: الجهد المطبق على المعالج.

I_{Leak} : تيار التسريب. [6]

4. بيئة التطوير المستخدمة في البحث

تم استخدام برنامج الماتلاب من أجل بناء النموذج الرياضي لخوارزمية الجدولة والمهام والمعالج متعدد النوى، بالإضافة إلى معادلات استهلاك الطاقة، وقد تم التوجه لاستخدام برنامج الماتلاب لما يتمتع به من ميزات عديدة ومكتبات برمجية ضخمة تضم العديد من التوابع البرمجية التي تؤمن القيام بمختلف العمليات الحسابية وعمليات النمذجة والمحاكاة بأسلوب مبسط وكفاءة عالية.

النتائج والمناقشة:

تمت دراسة ثلاثة سيناريوهات مختلفة تتضمن العمل وفق ثلاثة أحمال مختلفة للمعالج هي:

1. حالة الحمل الخفيف Light Load؛ وفيها تكون نسبة استخدام المعالج من قبل المهام Utilization تتراوح بين (0.2 – 0.5).

2. حالة الحمل العالي High Load؛ وفيها تكون نسبة استخدام المعالج من قبل المهام Utilization تتراوح بين (0.6 – 0.9).

3. حالة الحمل الزائد Over Load؛ وفيها تكون نسبة استخدام المعالج من قبل المهام Utilization تتراوح بين (1 – 1.3).

تم العمل على منصة تحتوي معالج بنواتين متطابقتين، وتمت مقارنة الأداء بين عمل الخوارزمية LLREF وفق تردد المعالج الأساسي (2.5 GHz) وبين عملها وفق قيمة للتردد يتم اختيارها في المجال (0.8 – 2.5 GHz) باستخدام الخوارزمية التطورية (PSO (Particle Swarm Optimization)، والتي تقوم بإيجاد القيمة المثلى لتردد عمل المعالج بحيث نحصل على أقل عدد ممكن من المهام التي تجاوزت قيدها الزمني – إن وُجدت – وبالتالي فإن أي تخفيض في قيمة التردد سوف يؤدي إلى تقليل في استهلاك الطاقة.

ويبين الجدول (1) بارامترات المعالج.

الجدول (1) بارامترات المعالج

عدد النوى	2
حجم L1 Cache	KB 32
حجم L2 Cache	KB 512
Line Size	Byte 16

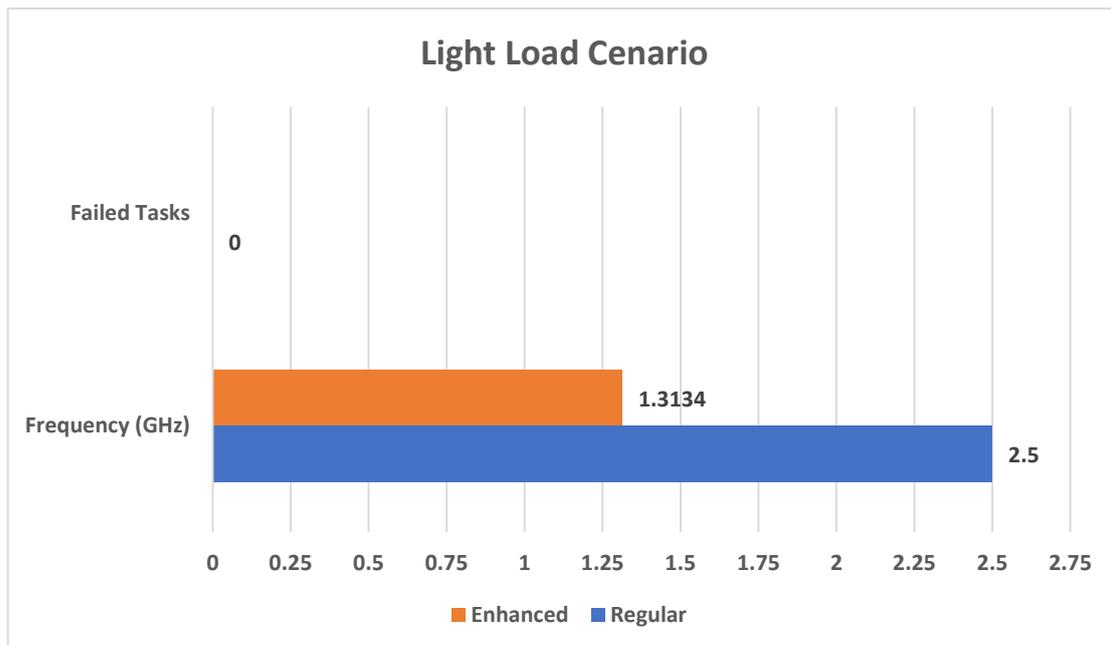
1. سيناريو الحمل الخفيف Light Load Scenario

تم في هذا السيناريو مقارنة الأداء بين عمل الخوارزمية LLREF قبل وبعد التعديل المقترح، ويبين الجدول (2) بارامترات المحاكاة.

الجدول (2) بارامترات سيناريو الحمل الخفيف

عدد المهام الأصغر	20 مهمة
عدد المهام الأكبر	35 مهمة
عدد تكرارات المحاكاة	30 تكرار
نوع الحمل	خفيف (0.2 – 0.5)

ويبين الشكل (2) نتائج المقارنة لسيناريو الحمل الخفيف



الشكل (2) نتائج سيناريو الحمل الخفيف

نستنتج من ملاحظة النتائج الواردة في سيناريو الحمل الخفيف أنه لا يوجد مهام تجاوزت قيدها الزمني وهذا أمر طبيعي كون الحمل خفيف، كما نلاحظ أنه يمكن إجراء عملية الجدولة والمحافظة على نفس الأداء عن طريق تقليل التردد إلى قيمة (GHZ 1.3434) وبالتالي سوف ينخفض استهلاك الطاقة بنسبة مئوية تبلغ (47.46%).

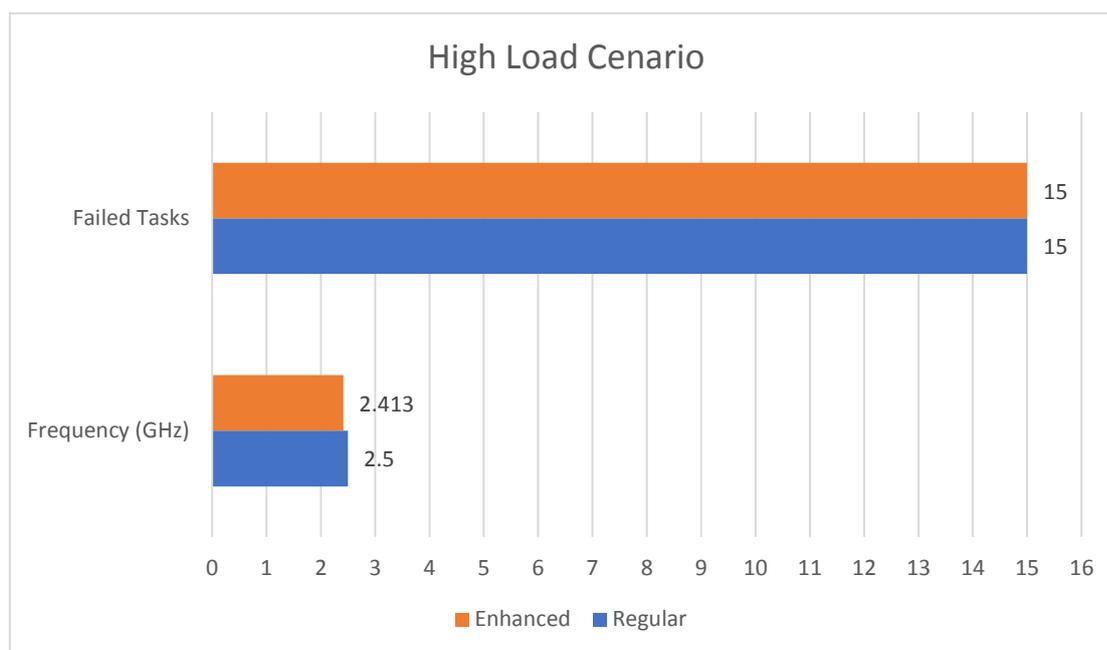
2. سيناريو الحمل العالي High Load Scenario

تم في هذا السيناريو مقارنة الأداء بين عمل الخوارزمية LLREF قبل وبعد التعديل المقترح، ويبين الجدول (3) بارامترات المحاكاة.

الجدول (3) بارامترات سيناريو الحمل العالي

عدد المهام الأصغر	30 مهمة
عدد المهام الأكبر	45 مهمة
عدد تكرارات المحاكاة	30 تكرار
نوع الحمل	عالي (0.6 – 0.9)

ويبين الشكل (3) نتائج المقارنة لسيناريو الحمل العالي



الشكل (3) نتائج سيناريو الحمل العالي

نستنتج من ملاحظة النتائج الواردة في سيناريو الحمل العالي أنه يوجد حوالي (15) مهمة تجاوزت قيدها الزمني وهذا أمر طبيعي كون الحمل عالي، كما نلاحظ أنه يمكن إجراء عملية الجدولة والمحافظة على نفس الأداء عن طريق تقليل التردد إلى قيمة (GHZ 2.4130) وبالتالي سوف ينخفض استهلاك الطاقة بنسبة مئوية تبلغ (3.4798%).

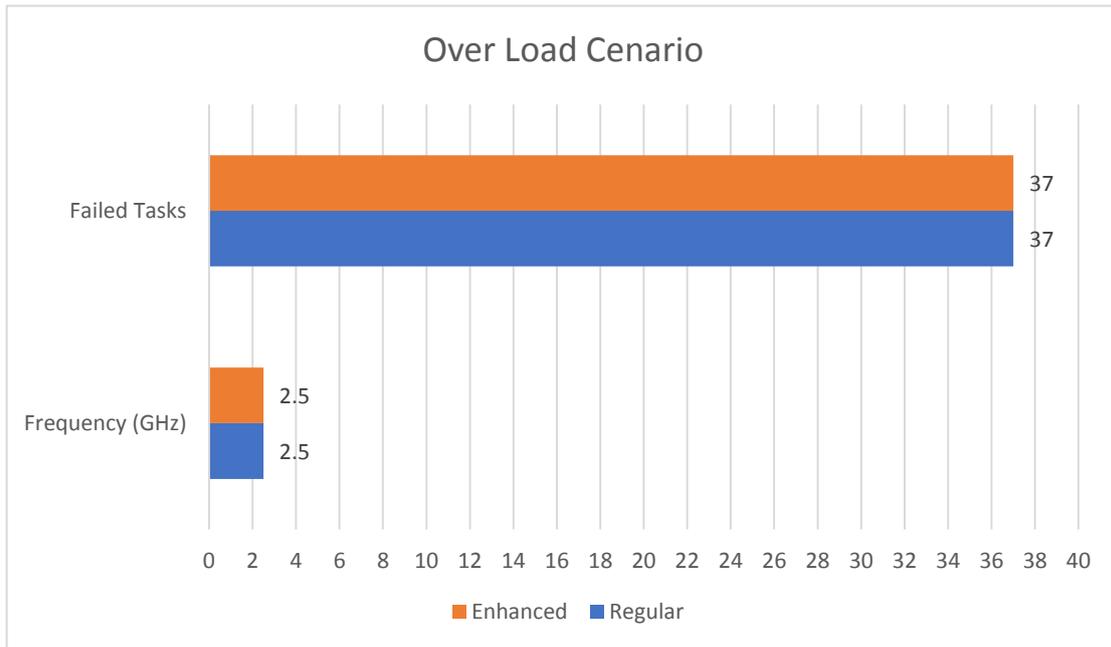
3. سيناريو الحمل الزائد Over Load Scenario

تم في هذا السيناريو مقارنة الأداء بين عمل الخوارزمية LLREF قبل وبعد التعديل المقترح، ويبين الجدول (4) بارامترات المحاكاة.

الجدول (4) بارامترات سيناريو الحمل العالي

عدد المهام الأصغر	40 مهمة
عدد المهام الأكبر	55 مهمة
عدد تكرارات المحاكاة	30 تكرار
نوع الحمل	زائد (1 - 1.3)

وبين الشكل (4) نتائج المقارنة لسيناريو الحمل الزائد



الشكل (4) نتائج سيناريو الحمل الزائد

نستنتج من ملاحظة النتائج الواردة في سيناريو الحمل الزائد أنه يوجد حوالي (37) مهمة تجاوزت قيدها الزمني وهذا أمر طبيعي كون الحمل زائد، كما نلاحظ أن الطريقة المقترحة أعطت نفس التردد الأساسي للمعالج (2.5 GHz) وبالتالي لا يوجد توفير لاستهلاك الطاقة في هذا السيناريو.

الاستنتاجات والتوصيات:

- يمكن من خلال الاطلاع على نتائج السيناريوهات الواردة في هذا البحث استنتاج مجموعة من النقاط نجملها فيما يلي:
1. نلاحظ أن الطريقة المقترحة أدت إلى توفير كبير في استهلاك الطاقة عندما كان الحمل خفيفاً (47.46%)، ومع ازدياد الحمل يقل توفير الطاقة حيث تصل النسبة إلى (3.48%) في حالة الحمل العالي، وهذا سبب منطقي كونه مع ازدياد الحمل يتطلب المعالج طاقة إضافية للعمل.
 2. نلاحظ أن الطريقة المقترحة تعطينا أفضل أداء لعمل المعالج من حيث مستوى تردد المعالج وذلك بحسب حمل المعالج.

3. لا يمكن توفير استهلاك الطاقة في حال الحمل الزائد، بسبب أن كمية المعالجة التي تتطلبها المهام تفوق نسبة استخدام المعالج.
4. يمكن الاستفادة من كمية الطاقة التي تم توفيرها في تنفيذ مهام إضافية وبالتالي زيادة كفاءة وديمومة نظام الزمن الحقيقي.

References:

1. Irraivan Elamvazuthi, "Electrical power consumption monitoring using a real-time system", Universiti Teknologi PETRONAS, (2021)
2. Radhakrishna Naik at el, "Periodic and Aperiodic Real - Time Task Scheduling Algorithms Simulator ", International Journal of Pure and Applied Mathematics, Volume 118 No. 20 ,2681-2687 (2018).
3. Hyeongboo BAEK, "Real-Time Scheduling for Preventing Information Leakage with Preemption Overheads ", Advances in Electrical and Computer Engineering, Volume 17, Number 2, (2017).
4. Schoeberl, M., Pezzarossa, "A multicore processor for time-critical applications ", Technical University of Denmark , (2018).
5. Charu Rani, Mrs. Manju Godara , " Real Time System Scheduling Algorithms & Fault Tolerance " , International Journal of Advanced Research in Computer and Communication Engineering , Vol. 4, Issue 7, July (2015).
6. Sri Raj Pradhan, Sital Sharma, Debanjan Konar, "A Comparative Study on Dynamic Scheduling of Real-Time Tasks in Multiprocessor System using Genetic Algorithms " , International Journal of Computer Applications , Volume 120 – No.20, June (2015).
7. Christine Rochange, "Parallel Real-Time Tasks, as Viewed by WCET Analysis and Task Scheduling Approaches " , University of Toulouse, Toulouse, France (2016).
8. Fernanda F. Peronaglio and Aleardo Manacero, "MODELING REAL-TIME SCHEDULERS FOR USE IN SIMULATIONS THROUGH A GRAPHICAL INTERFACE", Dept of Computer Science and Statistics São Paulo State University – UNESP, (2017).
9. Manar Qamheih, "Scheduling of Parallel Real-time DAG Tasks on Multiprocessor Systems", University of Paris-est (2015).
10. Kushal Anjaria, Arun Mishra, "Thread scheduling using ant colony optimization: An intelligent scheduling approach towards minimal information leakage", Department of Computer Science & Engineering, DIAT, Pune, India, (2017).
11. Gokul Sidarth Thirunavukkarasu*, and Ragil Krishna, "Scheduling Algorithm for Real-Time Embedded Control Systems Using Arduino Board", Deakin University, School of Engineering, Waurn ponds, Australia, (2017).
12. Amjad Mahmood , Salman A. Khan, "Energy-Aware Real-Time Task Scheduling in Multiprocessor Systems Using a Hybrid Genetic Algorithm", University of Bahrain, (2017).
13. Linlin Tanga, Kaiqiang Ma, Zuohua Li, "A New Scheduling Algorithm Based on Ant Colony Algorithm and Cloud Load Balancing", Harbin Institute of Technology Shenzhen Graduate School Shenzhen, China, (2016).
14. Khaldoun Faqi, Dr. Muhammad Hijazia, "Improving the performance of real-time systems by optimizing task scheduling on a multi-core processor", PhD thesis in the Department of Computer Engineering and Automatic Control, Tishreen University, Lattakia, Syria (2019).