نموذج رياضي مستمر للتسامح مع الأعطال في التطبيقات المتوازية

د. سمير جعفر

قسم الرياضيات – كلية العلوم – جامعة دمشق – سوريا

الملخص

نقدم في هذا البحث نموذج رياضي مستمر للحصول على الحل الأمثل للمشكلة الناتجة عن إضافة آلية للتسامح مع الأعطال في بيئات التنفيذ التفرعية والموزعة عالية الآداء وهي مشكلة التسوية بين الكلفة المضافة من آلية التسامح مع الأعطال وتأثير الأعطال على بيئة التنفيذ وبالتالي على زمن انتهاء تنفيذ التطبيق المتوازي. طريقة التسامح مع الأعطال المدروسة هي آلية تخزين/استرجاع متزامن والدراسة المقترحة تعتمد على نمذجة عشوائية مستمرة لمختلف قيود الأداء للتطبيق المتوازي المنفذ على بنية تفرعية موزعة.

الكلمات المفتاحية: التطبيقات المتوازية، البنى التفرعية الموزعة، التسامح مع الأعطال، الاحتمالات، نظرية الموثوقية، النمذجة الرياضية.

A continuous mathematical model of fault tolerance mechanisms for parallel applications

Dr. Samir Jafar

Department of Mathematics – Faculty of Science – Damascus University – Syria

Abstract

In this paper, we introduce a continuous mathematical model to optimize the compromise between the overhead of fault tolerance mechanism and the faults impacts in the environment of execution. The fault tolerance mechanism considered in this research is a coordinated checkpoint/recovery mechanism and the study based on stochastic model of different performance critics of parallel application on parallel and distributed environment.

Keywords: parallel applications, parallel and distributed environment, probability, reliability theory, mathematical modeling.

1. المقدمة:

إن الحاجة المستمرة والمتزايدة لإجراء كميات كبيرة من الحساب في مختلف المجالات والتخصصات العلمية أدت إلى تطور حقيقي في بنى (بيئات) التنفيذ التفرعية والموزعة. على سبيل المثال، عام 1997 البنية التفرع الأسرع التي احتلت المرتبة الأولى في TOP500 كانت مكونة من 100 معالج فقط [1]، أما الآن فالتطبيقات المتوازية من نوع HPC (High Performance Computing) تنفذ على بنى تحوي عشرات الآلاف من المعالجات [2].

هذه الزيادة الملحوظة في عدد المعالجات المشاركة في التنفيذ تفرض العديد من التحديات العلمية وبالأخص فيما يتعلق بموثوقية وسلامة تشغيل هذه البنى. جميع الدراسات الاحصائية [5][4][3] التي اجريت على هذه البنى اظهرت أن الزمن الوسطي بين الأعطال (MTTF) يتناسب عكساً مع عدد المعالجات المشاركة في التنفيذ وبالنتيجة فإن القدرة الحسابية للبنى التفرعية والموزعة الحالية والمستقبلية ستزداد أكثر فأكثر من جهة وستكون غير متوفرة وغير قابلة للإستخدام بسبب تزايد الأعطال الناتجة عن تزايد المعالجات المكونة لها من جهة أخرى.

بالتالي لابد من إضافة آلية للتسامح مع الأعطال [10] من أجل ضمان انتهاء تنفيذ التطبيقات المتوازية على هذه البنى المعرضة للأعطال باحتمال شبه أكيد. لإضافة مثل هذه الآلية نواجه مشكلتين اساسيتين يجب التعامل معهما:

1. مشكلة تصميم وتحقيق آلية التسامح مع الأعطال بما يتناسب مع طبيعة التطبيقات المنفذة ونوع الأعطال وطبيعة البنية التفرعية.
2. عندما نضيف الآلية المناسبة للتسامح مع الأعطال ونضعها قيد التشغيل فإنه لابد من إيجاد حل وسط (تسوية) بين الكلفة المضافة على تنفيذ التطبيق جراء بسبب هذه الآلية من جهة وآثار الأعطال على كلفة تنفيذ التطبيق من جهة أخرى.

2. هدف البحث:

هدفنا في هذا البحث حل المشكلة الثانية المطروحة في الفقرة السابقة حيث عن طريق بناء نموذج رياضي يمكننا من إيجاد الحل الأمثل لهذه المسألة.

بفرض أن آلية التسامح مع الأعطال المراد إضافتها هي طريقة التخزين/الاسترجاع المتزامن للتطبيق، نهدف إلى بناء نموذج عشوائي مستمر نستطيع من خلاله الإجابة على السؤال التالي:

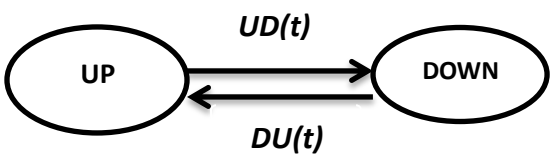
ما هو التردد الزمني الأنسب لإنشاء نقاط الإستعادة للتطبيق المتوازي المنفذ على بنية تفرعية متوازية من أجل تقليل زمن انتهاء تنفيذ هذا التطبيق؟

3. النمذجة الرياضية للأعطال:

من أجل تصميم استراتيجات التسامح مع الأعطال و مقارنة التحسينات التي توفرها هذه الاليات، لابد من نمذجة حصول الأعطال خلال التنفيذ ونمذجة المدة اللازمة لإصلاح هذه الأعطال أو تجاوزها

3.1. نمذجة العطل:

إن حالة المعالج (وحدة الحساب) تتناوب بين حالتين حسب سلسلة ماركوف [6] كما هو موضح في الشكل 1 التالي:



الشكل 1: سلسلة ماركوف المخصصة لمعالج أو لوحدة حساب

حيث نلاحظ أن حالة وحدة الحساب تتناوب بين حالتين: الأولى هي الحالة متوفر (UP) وفيها تكون وحدة الحساب متوفرة لتنفيذ البرنامج خلال UD وحدة زمن، في نهاية هذه المدة الزمنية يحصل عطل وتنتقل وحدة الحساب إلى الحالة الثانية وهي الحالة غير متوفر (DOWN) وتبقى فيها لمدة زمنية DU. يمكن تصور DU على أنه الزمن اللازم لإصلاح الجزء المعطل من وحدة الحساب أو مدة إنقطاع الشبكة عنها.

المدتان الزمنيتان المنقضيتان في الحالة متوفر والحالة غير متوفر لهما مدة زمنية عشوائية ويمكن نمذجتهما بمتحولين عشوائيين منفصلين ولهما التوزيع الاحتمالي نفسه [7] . بفرض $F(t)$ و $f(t)$ دالتي التوزيع والكثافة الاحتمالية لحصول الأعطال وبفرض أن $DOWN(t)$ و $down(t)$ دالتي التوزيع والكثافة الاحتمالية لمدة عدم التوفر على الترتيب.

بالتعريف دالة التوزيع F هي دالة متزايدة بالنسبة للزمن t وقيمها في المحال [0,t]. رياضياً الدالة $F(t)$ تمثل احتمال أن تتعرض وحدة الحساب إلى عطل في المجال [0,t]. كذلك نعرف $\overline{F}(t) = 1 - F(t)$ باحتمال أن تتعرض وحدة الحساب إلى عطل في المجال [t,+∞]. ونعلم أن $F(t) = \int_0^t f(x)dx$.

إن الزمن الوسطي بين الأعطال $MTBF$ يعطى بالعلاقة:

$$MTBF = \int_0^\infty f(x)dx$$

والانتقال من الحالة متوفر إلى الحالة غير متوفر وبالعكس يحدث حسب معدل الأعطال من جهة ومعدل مدة عدم التوفر من جهة أخرى. سنرمز لتابع معدل الانتقال من الحالة متوفر إلى الحالة غير متوفر بالرمز $UD(t)$ ولتابع معدل الانتقال من الحالة غير متوفر إلى الحالة متوفر بالرمز $DU(t)$ وبالتالي يكون [7] :

$$F(t) = 1 - e^{-\int_0^t UD(u)du} = \int_0^\infty f(x)dx$$

3.2. توزيع الأعطال:

حسب نظرية الموثوقية [8] نجد أنه يوجد بشكل أساسي توزيعيين احتماليين للأعطال:

- التوزيع الأسي: وهو الأكثر انتشاراً كونه بسيط رياضياً ويعتبر القانون الوحيد الذي لا يحوي ذاكرة أي: $P\{U \geq t\} = P\{U \geq s + t \mid U \geq s\}$ ونعلم أن القانون الأسي له وسيط واحد λ ودالة التوزيع له هي $F(t) = 1 - e^{-\lambda t}$ ودالة الكثافة الاحتمالية هي $f(t) = \lambda e^{-\lambda t}$ و $UD(t) = \lambda$.
- توزيع ويبل: وهو قانون ذو شعبية كبيرة بسبب مرونته العالية له وسيطين: وسيط الشكل β ووسيط القياس α ودالة التوزيع هي

$$F(t) = 1 - e^{-(\alpha t)^{\beta}}$$

ودالة الكثافة الاحتمالية تعطى كمايلي:

$$f(t) = \beta\alpha(t\alpha)^{\beta-1}e^{-(\alpha t)^{\beta}}$$

و

$$UD(t) = \beta\alpha(\beta t)^{\beta-1}$$

ويستخدم توزيع ويبل في الكثير من دراسات فترة البقاء (life time) والتي يمكننا من خلالها التنبؤ حول فترة البقاء.

4. نمذجة آلية التسامح مع الأعطال:

سنقوم أولاً بدراسة أداء التطبيق المتوازي المنفذ على بنية تفرعية معرضة للأعطال بدون آلية تسامح مع الأعطال حيث سنستخدم هذا النموذج كمرجع لتحليل ومقارنة

التحسينات المضافة باستخدام آلية التسامح مع الأعطال، ثم ثانياً بدراسة الأداء بوجود آلية للتسامح مع الأعطال.

1.4 نمذجة بدون تسامح مع الأعطال:

نوصف في هذه الفقرة نموذج الأداء للثنائية المكونة من التطبيق وبيئة التنفيذ بهدف توضيح أثر الأعطال على أداء التطبيق في حال عدم وجود آلية للتسامح مع الأعطال.

نمذجة بيئة التنفيذ:

نهدف في هذا العمل بيئات الحساب التفرعية والمتوازية عالية الأداء. ضمن هذه البيئات يوصف التطبيق المتوازي بكمية العمل الذي يحتويه حيث نفترض أن التطبيق المتوازي مكون من w وحدة عمل وينفذ على مجموعة من المعالجات المتماثلة والمتجانسة (نعلم أنه في بيئات التنفيذ عالية الآداء هذه المجموعة من المعالجات تعتبر وحدة حساب واحدة) وبالتالي يمكن نمذجة وحدة الحساب بالعناصر الثلاثة التالية:

1. معدل معالجة التعليمات خلال وحدة الزمن ونرمز لها v وتحتوي على عدد المعالجات المشاركة في التنفيذ واستطاعة التنفيذ لكل منها.
2. معدل التوفر (أو معدل العطل) ونرمز له $UD(t)$ والذي يمثل المدة الزمنية التي تكون فيها وحدة الحساب متوفرة لتنفيذ التطبيق. هذا المعدل يمثل تركيب قوانين الأعطال لجميع المعالجات. حيث تجدر الإشارة إلى أن التطبيق المتوازي يتعرض لعطل إذا تعطل معالج واحد عل الأقل من معالجات وحدة الحساب.
3. معدل عدم التوفر، نرمز له $DU(t)$ ، ويمثل المدة الزمنية اللازمة لإصلاح المعالج المعطل من وحدة الحساب علماً أن التطبيق المتوازي يعمل إذا وفقط إذا كانت جميع معالجات وحدة الحساب متوفرة. كذلك في حال تعطل معالج ما من وحدة الحساب (بيئة التنفيذ) فإنها تعتبر غير متوفرة.

بالنتيجة، في حال عدم وجود أعطال فإن الزمن اللازم لإنهاء تنفيذ التطبيق المتوازي المكون من w وحدة عمل على بيئة تنفيذ بمعدل معالجة V هو كمية محدودة تعطى بالعلاقة: $C = w/V$. من جهة أخرى، خلال التنفيذ الواقعي للتطبيق هناك أعطال تصيب معالجات البيئة بالضرورة مما سيؤدي إلى زيادة زمن انهاء التطبيق.

في التطبيقات المتوازية، العطل سيؤدي إلى خسارة العمل المنجز منذ بداية التنفيذ ولنرمز بـ L لكمية العمل الضائع نتيجة عطل ما. نعلم أنه عندما يقطع تنفيذ التطبيق بسبب عطل، لا يستطيع استئناف التنفيذ إلا بعد مدة من الزمن ولتكن DU (الزمن اللازم لإصلاح او استبدال المعالج المعطل) وعندها يبدأ التطبيق تنفيذه من البداية.

إن هذه العملية المكونة من المراحل: تنفيذ، توقف، إعادة التنفيذ، سوف تتكرر طالما أنه لا يوجد مدة توفر أكبر من الزمن اللازم لإنهاء تنفيذ التطبيق بدون أي عطل يوقف التنفيذ. بفرض DU و UD متحوليين عشوائيين بحيث UD_i يمثل مدة التوفر قبل العطل رقم i و DU_i يمثل مدة عدم التوفر بعد العطل رقم i . ضمن هذه المعطيات وبعد العطل رقم i فإن كمية العمل الضائع هي تماماً UD_i و DU_i وحدة زمن مضافة إلى زمن انتهاء تنفيذ التطبيق. الآن بفرض S_w هو رقم المجال بين عطليين متتاليين بحيث يحقق أنه أكبر من w عندئذٍ يعطى زمن انتهاء التطبيق $T(w)$ بوجود أعطال بالعلاقة:

$$T(w) = \sum_{i=1}^{S_w-1}(UD_i + DU_i) + w \qquad (1)$$

بالإعتماد على هذه العلاقة، سنقوم في الفقرات التالية بتقدير التوقع الرياضي لزمن انتهاء تنقيذ التطبيق وكذلك التوزيع الاحتمالي لعدد الأعطال والتوزيع الاحتمالي لمدة عدم التوفر ومقدار عمل التطبيق الأساسي المطلوب إنجازه.

التوزيع الاحتمالي لعدد مرات إعادة التنفيذ:

ليكن S_w عدد مرات إعادة التنفيذ اللازم من أجل انهاء تنفيذ w وحدة عمل (العمل المطلوب انجازه). إن الحدث $\{S_w = s\}$ يكافئ الحدث

$\{U_1 < w, U_2 < w, \dots, U_{s-1} < w, U_s > w\}$ وبالتالي S_w هو متحول عشوائي منقطع له التوزيع الهندسي بالوسيط q_w حيث:

$$q_w = P\{U > w\} = \bar{F}\,(w)$$

وبالتالي فإن التوزيع الاحتمالي والكثافة الاحتمالية لعدد مرات إعادة تنفيذ التطبيق المتوازي يعبر عنهما كما يلي:

$$P\{S_w = s\} = (1 - q_w)^{s-1} q_w \quad , \quad P\{S_w \le s\} = 1 - (1 - q_w)^s$$

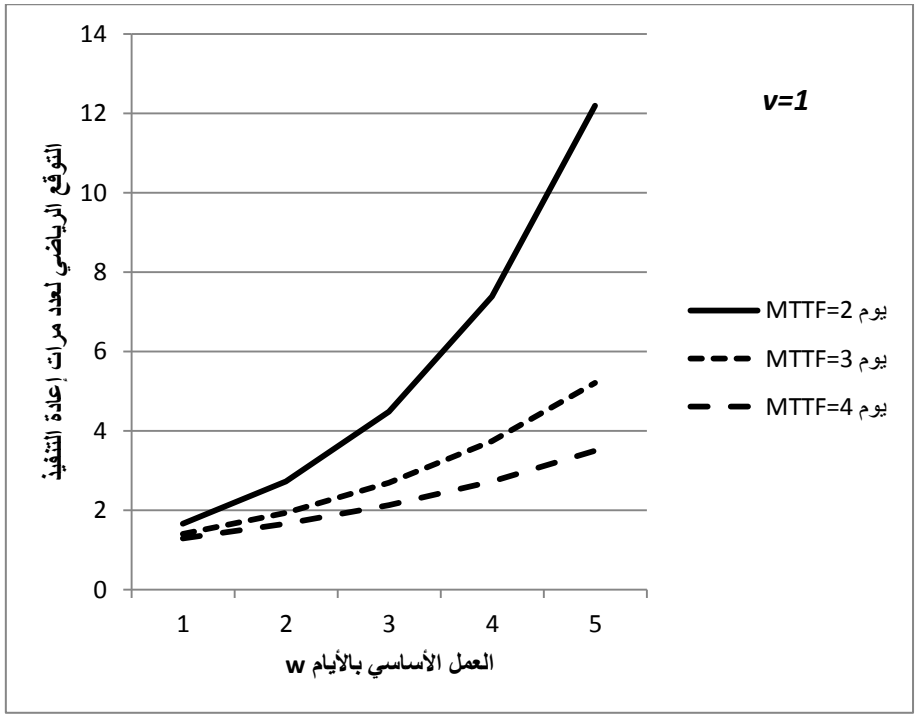
بالإعتماد على التوزيع الاحتمالي للمتغير S_w يمكننا حساب جميع الفترات، على سبيل المثال التوقع الرياضي والتباين لعدد مرات إعادة التنفيذ:

$$E(S_w) = \frac{1}{q_w} \quad , \quad VAR(S_w) = \frac{1 - q_w}{q_w^2}$$

والآن بفرض أن التوزيع الاحتمالي لحصول الأعطال هو التوزيع الأسي عندئذٍ يكون التوقع الرياضي والتباين لعدد مرات إعادة التنفيذ كما يلي:

$$E(S_w) = e^{\lambda w} \quad , \quad VAR(S_w) = e^{-2\lambda w} - e^{-\lambda w}$$

باستخدام هذه العلاقات نستطيع رؤوية تباين التوقع الرياضي لعدد مرات إعادة التنفيذ بدلالة كمية العمل الأساسي للتطبيق من أجل قيم مختلفة لمعدل الأعطال كما هو موضح في الشكل 2 التالي:



الشكل 2: التوقع الرياضي لعدد مرات إعادة التنفيذ

التوقع الرياضي لزمن انتهاء التنفيذ:

بالإعتماد على العلاقة (1) التي تعبر عن زمن انتهاء تنفيذ التطبيق بشكل عام، فإن التوقع الرياضي لزمن انتهاء التنفيذ يحسب كما يلي:

$$E\big(T(w)\big) = E\left(\left(\sum_{i=1}^{s_w-1} UD_i + DU_i\right) + w\right)$$

$$= E\left(\left(\sum_{i=1}^{s_w} UD_i + DU_i\right) + w - UD_{s_w} - DU_{s_w}\right)$$

$$= E\big(\textstyle\sum_{i=1}^{s_w} UD_i + DU_i\big) + w - E(UD_{s_w}) - E(DU_{s_w}) \qquad (2)$$

وبما أن UD_{s_w} هو متحول عشوائي له نفس التوزيع الاحتمالي للمتحول UD المشروط بالحدث $\{UD > w\}$ فهذا يعني أن

$$P\{UD_{s_w} = u\} = P\{\, UD = u \mid UD > w\}$$

وبالتالي فإن التوقع الرياضي للمتحول العشوائي UD_{s_w} يعطى بالعلاقة:

$$E(UD_{s_w}) = \frac{1}{\bar{F}(w)} \int_w^{\infty} x f(x) dx \qquad (3)$$

كذلك المتحول العشوائي DU_{s_w} والمتحول DU لهما التوزيع الاحتمالي نفسه كون DU مستقل تماماً عن S_w وبالتالي

$$E(DU_{s_w}) = E(DU)$$

وبحسب المرجع [9] نجد:

$$E(\sum_{i=1}^{S_w}(UD_i + DU_i) = E(S_w)(E(UD) + E(DU)) \qquad (4)$$

بتعويض (3) و (4) في (2) نجد:

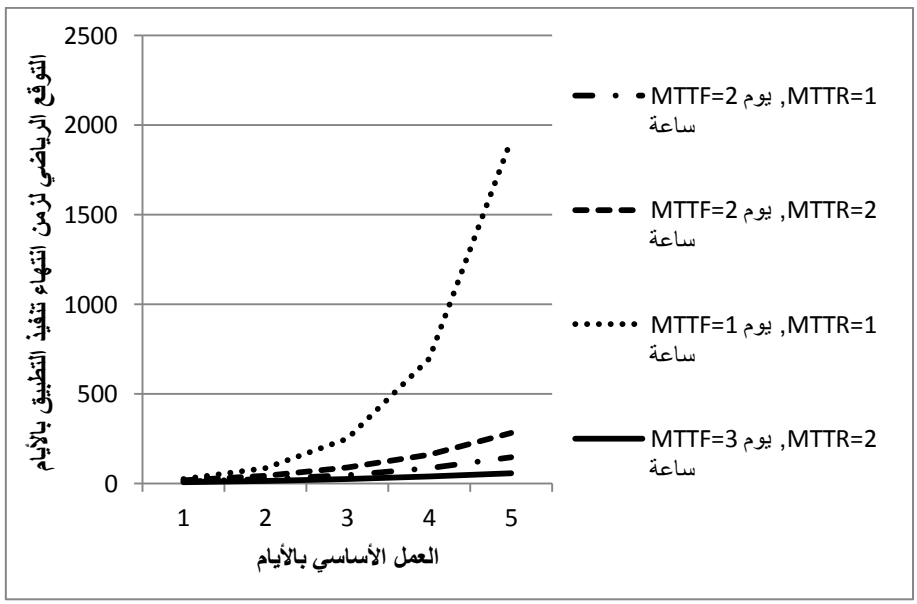
$$E\big(T(w)\big) = \frac{1}{\bar{F}(w)}(F(w)E(DU) + w\bar{F}(w) + \int_w^{\infty} x f(x) dx)$$

$$= \frac{1}{\bar{F}(w)}(F(w)E(DU) + w\bar{F}(w) - \int_0^{w} (1 - \bar{F}(x)) dx)$$

ومنه: $$E\big(T(w)\big) = \frac{1}{\bar{F}(w)}(F(w)E(DU) + \int_0^{w} \bar{F}(x) dx) \qquad (5)$$

لنفرض الآن أن التوزيع الاحتمالي للأعطال هو التوزيع الأسي بالوسيط λ والتوزيع الاحتمالي لمدة عدم التوفر هو التوزيع الأسي كذلك بالوسيط γ ، عندئذٍ يكون التوقع الرياضي لزمن انتهاء تنفيذ التطبيق المتوازي حسب العلاقة (5) كما يلي:

$$E\big(T(w)\big) = \left(e^{\lambda t} - 1\right)\left(\frac{1}{\lambda} + \frac{1}{\gamma}\right) \qquad (6)$$



الشكل 3: تباين التوقع الرياضي لزمن انتهاء تنفيذ التطبيق المتوازي بدلالة مقدار العمل الأساسي w ومعدل العطل (MTTF) ومدة عدم التوفر (MTTR)

باستخدام العلاقة (6) نستطيع مشاهدة تباين التوقع الرياضي لزمن انتهاء التنفيذ كما هو موضح في الشكل 3 حيث نلاحظ بشكل واضح أن التوقع الرياضي لزمن انتهاء التنفيذ يتزايد بشكل أسي في حالة التوزيع الأسي للأعطال ولمدة الإصلاح وهذا التزايد يعود

للمقدار $\frac{F(w)}{\bar{F}(w)}$ وبالتالي الطريقة الأسهل التي تسمح بتخفيض التزايد الأسي إلى تزايد خطي هي تقسيم التطبيق المتوازي إلى أجزاء منفصلة أي تقسيمه إلى مهام مستقلة عن بعضها البعض.

2.4 نمذجة بوجود التسامح مع الأعطال:

بعد أن قمنا بنمذجة الأداء بدون آلية للتسامح مع الأعطال سنقوم في هذه الفقرة بربط النموذج السابق مع آلية للتسامح مع الأعطال من نوع تخزين/استرجاع متزامن وذلك سيسمح لنا بتقدير مساهمة هذه الآلية في تحسين أداء التنفيذ.

نمذجة آلية التخزين/الاسترجاع المتزامن:

بالإعتماد على مفهوم طريقة التخزين/الاسترجاع المتزامن [10] المستخدمة في هذا العمل كآلية للتسامح مع الأعطال، سنقدم في هذه الفقرة التوصيف الرياضي لنمذجة هذه الآلية.

بفرض لدينا تطبيق متوازي يحوي على w وحدة عمل (وبفرض أن $v = 1$)، إن إضافة نقاط إستعادة إلى التطبيق يعتمد على تقسيم العمل إلى n جزء مستقل تنفذ بالترتيب. تقنياً وفي إطار النمذجة المستمرة، هذا سيؤدي أن تنفيذ التطبيق يمكن أن يقاطع في أي لحظة من أجل إنجاز نقطة الإستعادة. إذاً الحدود τ_i بين أجزاء التطبيق (n جزء) ستكون معروفة وموصفة حسب استراتيجية جدولة نقاط الإستعادة المعتمدة ولنرمز Π لهذه الاستراتيجية حيث يمكننا توصيفها رياضياً بطريقتين متكافئتين:

1. بفرض $\Pi \in \mathbb{R}_+^n$ هي مجموعة الاستراتيجيات الصحيحة لجدولة نقاط الاستعادة، عندئذٍ الاستراتيجية $\pi \in \Pi$ التي تحوي n نقطة استعادة تحقق بالضرورة الشروط التالية:

$$\pi \in \Pi \ \ iff \ \begin{cases} \forall(\tau_i, \tau_j) \in \pi, i < j \ \Rightarrow \tau_i < \tau_j \\ \tau_n = w \end{cases} \qquad (7)$$

2. المجالات بين نقاط الاستعادة يرمز لها بالمتتالية $\{\rho_i\}_{i\leq n}$

والطريقتان السابقتان متكافئتان بملاحظة أن:

$$\forall i: 1 \leq i \leq n, \tau_i - \tau_j = \rho_i \,, \tau_0 = 0 \; and \; \sum_{i=1}^{n} \rho_i = \tau_n$$

التوقع الرياضي لزمن انتهاء التنفيذ بوجود آلية تخزين/استرجاع متزامن:

إن إضافة نقاط إستعادة إلى التطبيق المتوازي يكون حسب استراتيجية جدولة لهذه النقاط تؤدي إلى تجزئة تنفيذ التطبيق إلى n جزء. تنفذ هذه الأجزاء واحداً تلو الآخر بشكل تسلسلي. في حال حدوث عطل ما، يضيع فقط الجزء قيد التنفيذ. في هذه الحالة، التوقع الرياضي لزمن انتهاء التنفيذ لكل جزء من أجزاء التطبيق يكون مستقل عن الآخر ويحسب من العلاقة (5) وبالتالي فإن التوقع الرياضي لزمن انتهاء كامل التطبيق هو مجموع التوقعات الرياضية لهذه الأجزاء.

من جهة أخرى نلاحظ أن العلاقة (5) تعطي التوقع الرياضي لزمن الانتهاء من w وحدة عمل علماَ أن التنفيذ يبدأ في كل مرة من لحظة البدء أي $t = 0$. بتعبير آخر عندما يكون عمر التنفيذ في وحدة الحساب يساوي 0 وحدة زمن. ولكن من الممكن أن تكون عدة أجزاء قد انهت تنفيذها عند حدوث العطل. بالتالي من أجل جزء ما من العمل لا يبدأ من البداية يكون عمر التنفيذ في وحدة الحساب يساوي الزمن الذي توجب من أجل تنفيذ الأجزاء السابقة مع نقاط الاستعادة الخاصة بكل منها، وبالتالي من أجل اعتماد العلاقة (5) في هذه الحالة يكفي أن نأخذ التوزيع الاحتمالي للعطل $F(x)$ مشروطاً بالحدث التالي:{ وحدة الحساب في الحالة متوفر منذ t وحدة زمن } ومنه يكون التوقع الرياضي لزمن انتهاء الجزء ρ_i مع نقطة الاستعادة الخاصة به c_{τ_i} معطى بالعلاقة:

$$E\left(T\left(\rho_i + c_{\tau_i}\right)\right) = \frac{1}{\bar{F}_{a_i}\left(\rho_i + c_{\tau_i}\right)}\left(F_{a_i}\left(\rho_i + c_{\tau_i}\right)E(DU) + \int_0^{\rho_i + c_{\tau_i}} \bar{F}_{a_i}(x)dx\right)$$

حيث $a_i = \tau_{i-1} + \sum_{j=1}^{i-1} c_{\tau_j}$ هو عمر تنفيذ وحدة الحساب علماً أن $a_1 = 0$.

و $F_{a_i}(x)$ التوزيع الاحتمالي الشرطي حيث:

$$F_{a_i}(x) = (F(x + a_i) - F(a_i))/\,\overline{F}(a_i)$$

وتجدر الإشارة إلى أنه يمكن استخدام العلاقة (5) كما هي في حال كان معدل الأعطال ثابت لأن التوزيع الأسي بدون ذاكرة. إذاً بالإعتماد على ماسبق يمكن التعبير عن التوقع الرياضي لزمن انتهاء التطبيق بوجود ألية تخزين/استرجاع متزامن باستراتيجية جدولة π بالعلاقة التالية:

$$E\big(T_\pi(w)\big) = \sum_{i=1}^{n} E\left(T\big(\tau_i - \tau_{i-1} + c_{\tau_i}\big)\right) = \sum_{i=1}^{n} E\big(T\big(\rho_i + c_{\tau_i}\big)\big) \qquad (8)$$

مسألة الحل الأمثل لجدولة نقاط الإستعادة:

بالإعتماد على العلاقة (8) التي تعبر عن التوقع الرياضي لزمن انتهاء تنفيذ التطبيق بدلالة استراتيجية جدولة نقاط الإستعادة π، يمكننا الآن صياغة مسألة الحل الأمثل من أجل حساب الاستراتيجية الأمثل لجدولة نقاط الإستعادة ولنرمز لها π^* والتي تقلل التوقع الرياضي لزمن انتهاء تنفيذ التطبيق.

لتكن $Z: \mathbb{R}^n \longrightarrow \mathbb{R}$ دالة الهدف و $\Pi \neq \emptyset$ مجموعة جزئية من $\mathbb{R}^n$ التي تحوي على حلول صحيحة للمسألة والتي تدعى أيضاً مجموعة القيود. عندئذٍ يمكن صياغة المسألة بالشكل الآتي:

$$\min_{\pi \in \mathbb{R}^n} Z(\pi) \qquad \text{بشرط } \pi \in \Pi \qquad (9)$$

حيث:

$$Z(\pi) = \sum_{i=1}^{n}\left(\frac{\left(F_{a_i}\left(\tau_i - \tau_{i-1} + c_{\tau_i}\right)E(DU) + \int_0^{\tau_i - \tau_{i-1} + c_{\tau_i}} \bar{F}_{a_i}(x)dx\right)}{\bar{F}_{a_i}\left(\tau_i - \tau_{i-1} + c_{\tau_i}\right)}\right)$$

ومجموعة القيود Π المعطاة بالعلاقة (7).

وهي مسألة حل أمثل يمكن حلها بإحدى الطرق المقترحة في بحوث العمليات.

4. الإستنتاجات والتوصيات:

قمنا في هذا العمل بدراسة أداء تطبيق متوازي بوجود الأعطال وبدون وجود آلية للتسامح مع الأعطال. هذه الدراسة أظهرت أن إضافة آليات للتسامح مع الأعطال هو أمر جوهري لا يمكن الإستغناء عنه حيث بدونها يتزايد التوقع الرياضي لزمن انتهاء تنفيذ التطبيق بشكل أسي بالنسبة لزيادة العمل أو لزيادة معدل الأعطال في حال كانت تتبع التوزيع الأسي. وكذلك تبين أن آلية للتسامح مع الأعطال من نوع تخزين/استرجاع هي حل فعال من أجل تخفيض هذا التزايد الأسي. مثل جميع آليات التسامح مع الأعطال، عند وضعها قيد التشغيل، تظهر لنا مشكلة جديدة وهي مشكلة إيجاد حل وسط او تسوية مقبولة بين كمية العمل الضائع بسبب الأعطال والكلفة الناتجة عن إنشاء وتخزين نقاط الإستعادة. لذلك قمنا بصياغة مسألة حل أمثل والهدف منها حساب استراتيجية جدولة لنقاط الإستعادة بحيث تقلل ما أمكن التوقع الرياضي لزمن انتهاء التنفيذ. نستنتج بوضوح من هذه الدراسة أنه في حال كان لدينا معدل ثابت للأعطال وكذلك كلفة ثابتة لنقاط الإستعادة فإن الإستراتيجية الأمثل لجدولة نقاط الإستعادة هي بالضرورة استراتيجية دورية، ولابد من حل المسألة (9) بالنسبة لباقي الحالات لتعين الإستراتيجية الأمثل.

5. الخاتمة:

قدمنا في هذا البحث عدة نماذج رياضية توصف كل من الأعطال وبيئة التنفيذ التفرعية وكذلك تنفيذ التطبيق المتوازي مع وبدون آلية للتسامح مع الأعطال. وتمكن من صياغة وتوصيف مسألة الحل الأمثل للمشكلة الناتجة عن إضافة آلية للتسامح مع الأعطال في بيئات التنفيذ التفرعية والموزعة عالية الآداء وهي مشكلة تسوية لإيجاد حل وسط بين الكلفة المضافة من آلية التسامح مع الأعطال وتأثير الأعطال على بيئة التنفيذ وبالتالي على زمن انتهاء تنفيذ التطبيق المتوازي.

6. المراجع

[1] Feitelson D.G,2005-The supercomputer industry in light of the top500 data Computing in Science Engineering,7(1):42-47

[2] Oldeld R.A.and all, 2007-Modeling the impact of checkpoints on next-generation systems, 24th IEEE conference on mass storage systems and technologies, pages30-46

[3] Cappello F., Geist A., Gropp B., Kale L., Kramer B. and Snir M., 2009-Toward exascale resilience, International Journal of High Performance Computing Applications, 23(4) :374

[4] Oliner A. and Stearley J, 2007- What supercomputers say : A study of five system logs, In 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks,pages 575-584

[5] Schroeder B. and Gibson G.A., 2010-A large-scale study of failures in high-performance computing systems,IEEE Transactions on Dependable and Secure Computing,7(4) :337-351

[6] Puterman M.L., 1994-Markov decision processes: Discrete stochastic dynamic programming, John Wiley & Sons, Inc. New York, USA.

[7] Barlow R.E., Proschan F. and Hunter L.C., 1996-Mathematical theory of reliability, SIAM.

[8] Osman H. and Tahar S., 2016-Formalized Probability Theory and Applications Using Theorem Proving, IGI Global, 1-298, doi:10.4018/978-1-4666-8315-0

[9] Tijms H. C., 2003-A First Course in Stochastic Models, John Wiley.

[10] Abdelhafidi, Z., Djoudi, M., Lagraa, N., and Yagoubi, M. B.,2015-FNB: Fast Non-Blocking Coordinated Checkpointing Protocol for Distributed Systems. Theory of Computing Systems, 57(2), 397-425