

# Algorithme de Naimi-Tréhel et tolérance aux pannes

Raimana BOUISSOU, Tom GIMENEZ, Léo RIZZON

**Ici, j'écris à la suite, sans mise en page, sans fichier séparé. Je le ferai ensuite.**

## 1 Introduction

Dans le cadre de notre unité d'enseignement intitulée Algorithmie Distribuée nous avons réalisé un projet qui consiste en l'implémentation d'une amélioration du modèle de tolérance de l'algorithme de Naimi-Tréhel .

Nous nous sommes basés sur un article co-écrit par Julien Sopena, Luciana Arantes, et Pierre Sens qui propose un algorithme d'exclusion mutuelle, tolérant aux fautes. Basé sur l'algorithme de Naimi-Tréhel , il apporte une nette amélioration du coût de recouvrement en terme de message.

## 2 Objectifs

L'objectif de ce projet était, d'une part, la compréhension poussée d'un algorithme d'exclusion mutuelle et l'étude d'un système tolérant aux pannes. Toutefois, il s'agissait assimiler les problèmes et défauts qu'il comportait pour bien appréhender l'amélioration proposée.

D'autre part, l'implémentation de ces modèles pour appliquer toute la théorie à un exemple concret bien que futile.

Notons également que l'étude s'est faite à partir d'un article en anglais, comme la majorité des articles scientifiques, ce qui constitue une expérience intéressante.

L'implémentation consiste donc en l'application de l'algorithme de Naimi-Tréhel à un certain nombre de sites, en réseaux, qui cherchent à accéder à une même ressource alors qu'il n'y qu'un accès possible à la fois.

## 3 Les algorithmes étudiés

Cette partie nous servira à apporter une explication, avec nos propres mots et exemples, des algorithmes que nous avons étudiés.

### 3.1 Algorithme de Naimi-Tréhel

L'algorithme de Naimi-Tréhel est un modèle permettant à plusieurs sites de fonctionner autour d'une ressource. A un moment donné on veut être assuré qu'un seul site sera en Section Critique. C'est un algorithme d'exclusion mutuelle fonctionnant sur le principe du jeton.

**Fonctionnement de l'algorithme** Le fonctionnement général repose sur la construction dynamique de deux arbres. Aucun site n'a de vue globale de ces arbres et ne connaît alors que son *last*, dernier possesseur connu du jeton par ce site et son *next* à qui ce site doit passer le jeton une fois sorti de S.C.

Ainsi, le cas échéant, un site sait vers qui demander le jeton et vers qui le transmettre une fois terminé.

#### Implémentation

### 3.2 Fonctionnement de l'algorithme

### 3.3 Implémentation

ors

### 3.4 L'extension de Naimi-Tréhel permettant la tolérance aux pannes

Dans sa version la plus simple, l'algorithme de Naimi-Tréhel n'assure pas la tolérance aux pannes. Une extension existe toutefois pour répondre à ce besoin.

**Explication du mécanisme** Dans des conditions réelles d'utilisation il existe toujours un risque de panne sur un ou plusieurs sites. Les modèles utilisant le principe du jeton posent de plus le problème de la perte de celui-ci. De plus, dans notre cas, la construction d'arbres virtuels dépourvue de vision globale engendre le risque de perversion complète de l'organisation des sites.

Ici, on utilisera, pour détecter une faute, un timeout. Il s'agit d'un timer que l'on considère comme étant un temps raisonnable d'attente avant de suspecter un problème. Notons que ce timeout est à adapter à la configuration : temps de transfert des messages, nombre de sites, temps de S.C.. Lors de la demande du jeton, un site attendra un certain timeout avant de lancer une procédure de vérification de défaillance. Ainsi il envoie un message spécifique à tous les autres sites afin de vérifier si son prédécesseur dans la file des *next*(alors inconnu) est opérationnel. Il attend alors à nouveau un certain timeout car si c'est bien son prédécesseur qui est crashé, il ne recevra pas de réponse. Dans le cas d'une réponse, ce site n'a alors plus de questions à se poser et ne change rien, on déduit en effet, s'il existe bien une panne, que c'est son prédécesseur fera alors le travail nécessaire. Dans le cas d'absence de réponse on peut avoir deux cas de figure. En effet, dans le pire des cas c'est le site possédant le jeton qui a eu une panne. Le jeton est alors perdu et ne peut être regeneré n'importe comment. Un nouveau message spécifique est alors envoyé à l'intention du possesseur du jeton. Si celui-ci répond, le site questionneur referra alors simplement sa demande de jeton pour palier à une éventuelle perte de message lors de la première. Par contre si celui-ci ne répond pas, une réinitialisation globale très coûteuse est effectuée. Une élection a lieu pour regenerer un jeton unique et toute l'organisation est perdue et doit être refaite. Toutes les demandes doivent être effectuées de nouveau.

## **Implémentation**

**Atouts et inconvénients de cette extension** Cette extension répond de manière sûre au problème de pannes. Néanmoins dans le cas de la perte du jeton on a affaire à une réinitialisation complète particulièrement peu efficace et coûteuse. Pire encore, on peut arriver à des coûts encore plus élevés dans certains cas où le jeton n'est pas perdu : on constate alors des procédures de recouvrement individuel concurrentes.

### **3.5 Explication du mécanisme**

### **3.6 Implémentation**

### **3.7 Atouts et inconvénients de cette extension**

### **3.8 Amélioration de la tolérance aux pannes**

Nous avons vu que, malgré qu'elle soit efficace, l'extension de l'algorithme de Naimi-Tréhel n'est pas efficiente. C'est dans une optique d'amélioration

de ce système que l'algorithme équitable tolérant aux fautes à été imaginé.

**Les différents mécanismes** Tout d'abord les sites possèdent désormais plus d'informations que dans l'algorithme précédant. Ils connaissent leur position dans la file de *next* mais également un certain nombre de leur prédécesseur. On voit une réponse directe aux problèmes posés par l'algorithme précédant. Là où un broadcast était nécessaire pour trouver son prédécesseur avant, aucun message n'est utile ici !

## **Implémentation**

### **3.9 Les différents mécanismes**

#### **3.10 Implémentation**

#### **3.11 L'algorithme de Naimi-Tréhel sur des exemples**

#### **3.12 Naimi-Tréhel sans extension**

#### **3.13 Naimi-Tréhel avec extension**

## **4 Le developpement**

Cette partie apportera quelques précisions sur la manière dont nous avons travaillé,

### **4.1 Choix techniques**

#### **4.1.1 Le langage de programmation**

Nous avons choisi le C comme langage de programmation. Plusieurs raisons ont motivé ce choix. D'une part nous avons une solide expérience avec ce langage. Nous trouvons facilement nos marques. De plus, avec le C, nous avons les clés dont nous avons besoin : réseau, traitement d'une ressource, multi-threading.

#### **4.1.2 L'architecture**

L'implémentation soulève un certain nombre de questions qui restent en retrait dans la partie théorique. Nous avons alors dû choisir une architecture cohérente avec nos besoins : réseau, travail en Section Critique.

Par exemple, dans la théorie, les sites restent à l'écoute de messages alors qu'ils sont en S.C. ce qui nécessite des actions non bloquantes. Ainsi, le choix du multi-threading s'est révélé idéal. Ainsi, on retrouve un schéma d'exécution comme présenté figure 1.

Tous les sites sont équivalant, ils sont tous écrit avec le même code.

Le choix du TCP vient principalement de notre expérience en réseau : nous sommes plus à l'aise avec ce protocole. On peut noter toutefois que TCP fournit un certain nombre de services qui ne doivent pas être utilisés dans notre projet. En effet, la détection de pannes est alors aisée avec le système en mode connecté qui surveille la perte de lien entre deux sites. Dans l'intérêt de notre projet nous n'avons bien sûr pas utilisé cette fonction.

## **4.2 L'implémentation**

### **4.2.1 L'implémentation des messages**

En chaîne de caractère, donner un exemple.

### **4.2.2 La ressource utilisée en Section Critique**

Traitement sur une images ? Si oui en dire un peu plus

### **4.2.3 Des variables globales pour plus de simplicité**

## **5 Conclusion**

### **5.1 Bilan technique**

Bilan par rapport aux objectifs. Trucs qui marchent pas, à améliorer...

### **5.2 Bilan personnel**

Ce que ça nous a apporté, intérêt, expérience...