COMPTE-RENDU

TP numéro 3 [LO22]

02 MAI 2016

Marion Chan-Renous-Legoubin & Laviolette Etienne

Chan-Renous-Legoubin Marion
Laviolette Etienne

Le 2 Mai 2016

Compte rendu TP3

Rappel des objectifs du TP :

L'objectif principal de ce TP numéro 3 de LO22 est d'évaluer les paramètres de sûreté de fonctionnement (fiabilité, disponibilité, etc.) d'un système informatique à l'aide de la méthode des graphes de Markov, en utilisant l'outil python "Fiabilipy". Ainsi, au cours de ce TP, nous avons étudié deux applications différentes d'un même système informatique : un serveur. Nous nous sommes attelé à étudier un serveur constitué tout d'abord de deux machines puis de trois machines dans des configurations différentes (série, redondance active, voteur 2/3).

I. Application n°1

Chan-Renous-Legoubin Marion
Laviolette Etienne

Le 2 Mai 2016

Compte rendu TP3

La première architecture à étudier est composée de deux machines A et B constituant un serveur informatique. Ces deux machines aux taux de défaillance ainsi qu'aux taux de réparation distincts sont disposés en redondance active. Nous avons modélisé ce système par la technique des graphes de Markov sur fiabilipy.

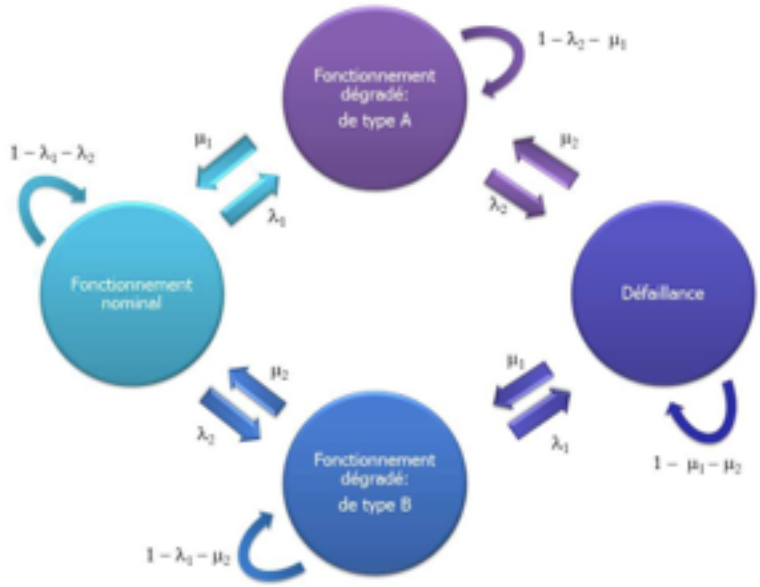
Il est à noter que les deux machines ont un rôle symétrique : elles effectuent les mêmes requêtes. On les nomme hétérogène puisqu'elles utilisent simplement des technologies différentes.

$\lambda_1 = 0.0001/\text{heure}$

$\mu_1 = 0.0011/\text{heure}$

$\lambda_2 = 0.0004/\text{heure}$

$\mu_2 = 0.0014/\text{heure}$

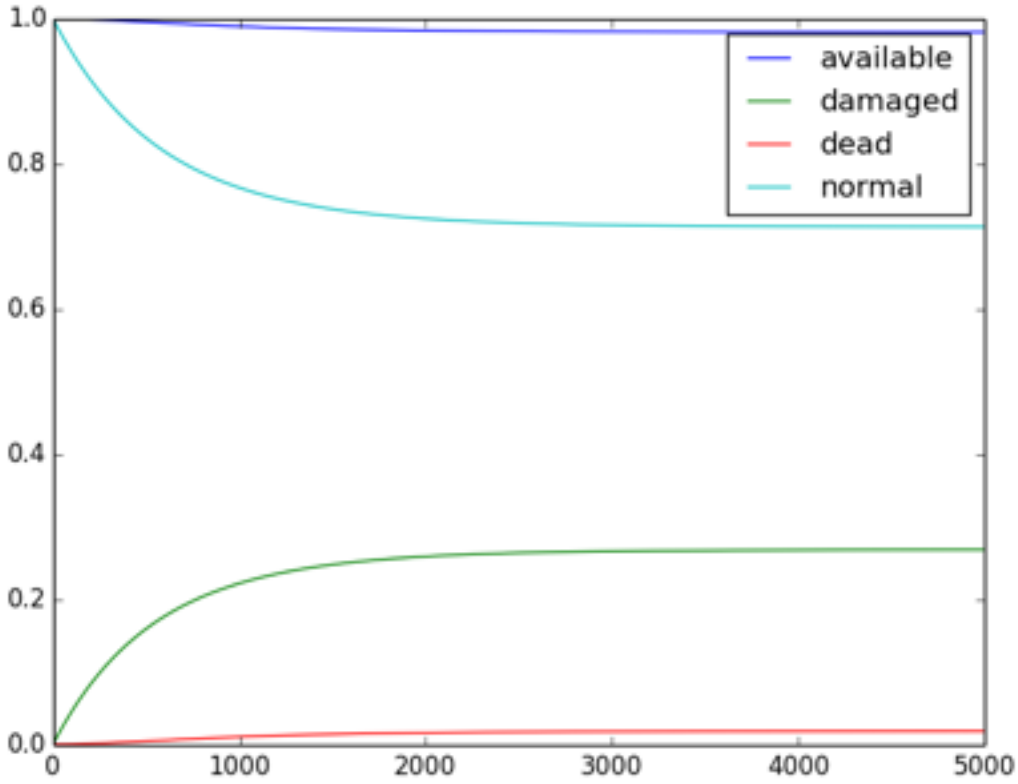


Graph de Markov du serveur informatique à deux machines

A l'aide des valeurs de taux de défaillance et de taux de réparation fournis pour les deux machines A et B et grâce à l'outil fiabilipy nous avons pu tracer les courbes de probabilité dans les états suivants au cours du temps :

- Fonctionnement nominal (normal)

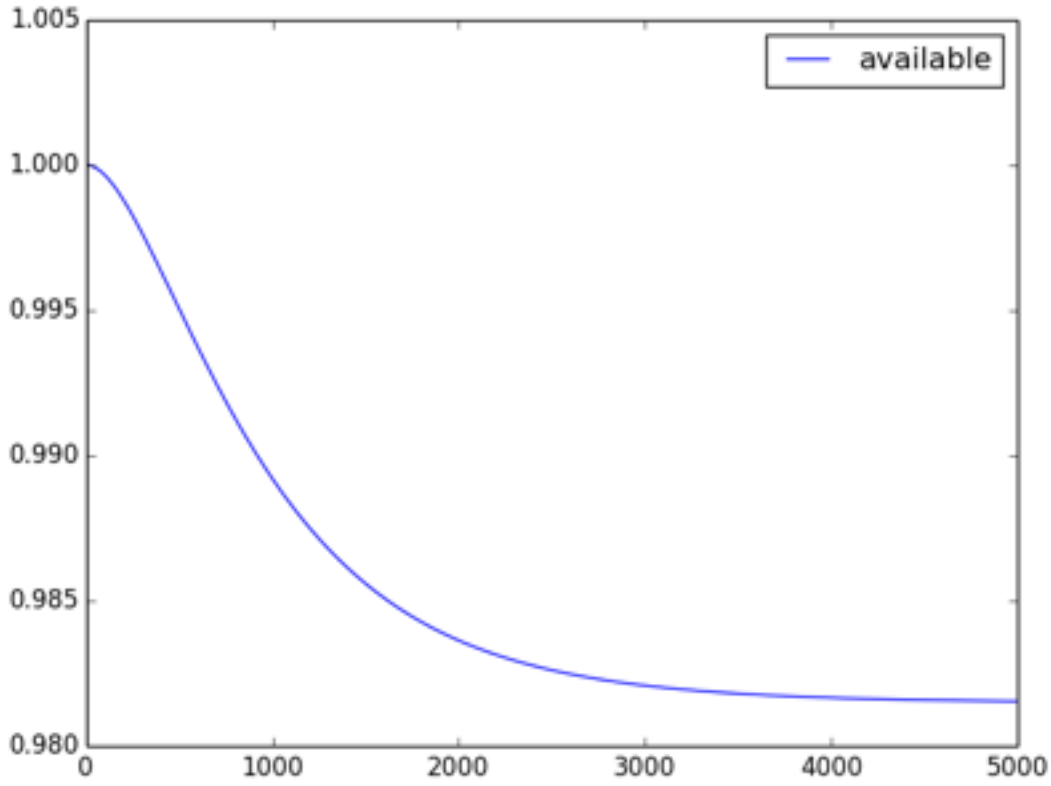
- Fonctionnement dégradé (damaged)
- Défaillance (dead)



Courbes de probabilité des différents états de notre application 1

<u>Nota</u> : Vous trouverez en annexe de notre rapport les courbes de probabilité des différents états dans des échelles plus exploitables.

Vous pouvez aussi retrouver sur notre capture d'écran la courbe de disponibilité de notre première application. Cependant celle-ci n'est pas exploitable avec l'échelle présentée. La voici présentée de manière plus explicite :



Courbes de disponibilité de notre application 1

La valeur asymptotique de la disponibilité de notre première application est de :

```
print(process.value(5000, available))
#0.981529664001
```

II. Application n°2

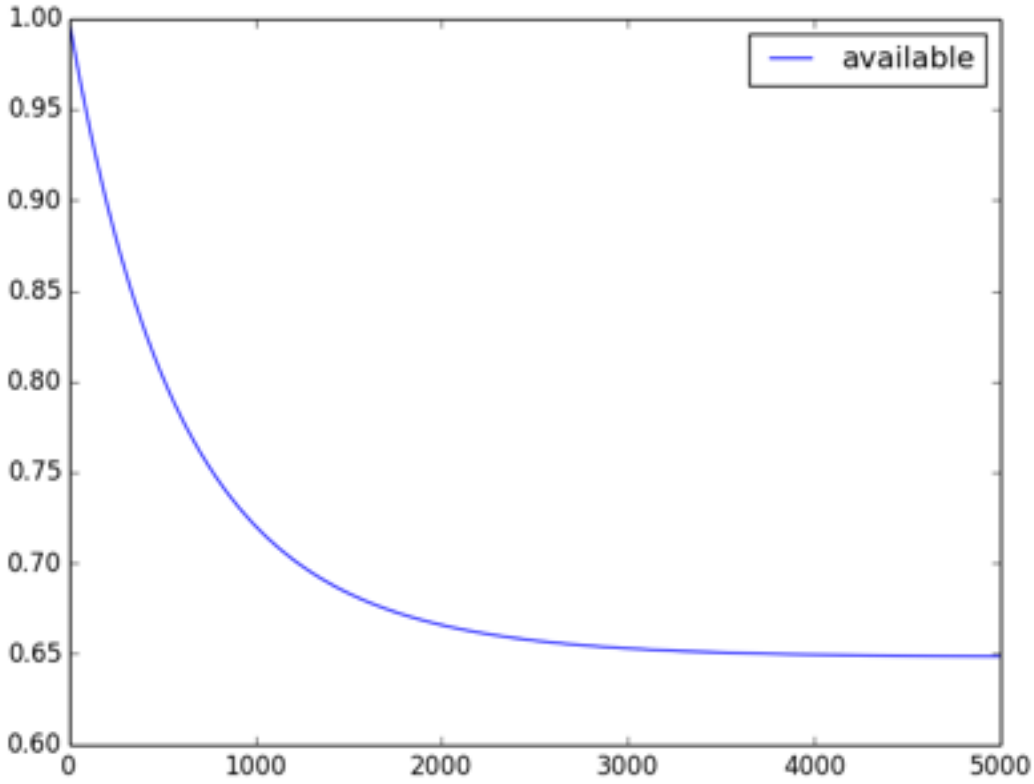
La seconde architecture à étudier est composée de trois machines A, B et C constituant un serveur informatique. Nous avons modélisé ce système par la technique des graphes de Markov sur fiabilipy.

Nous reprenons les mêmes valeurs de taux de défaillance et de taux de réparation que notre application numéro 1. Nous rajoutons une valeur de chacun de ces paramètres pour la troisième machine :

$\lambda_3 = 0.0001/\text{heure}$

$\mu_3 = 0.0010/\text{heure}$

La première version de notre application numéro 2 est constituée des 3 machines en série
Nous traçons la courbe de disponibilité pour ce système :

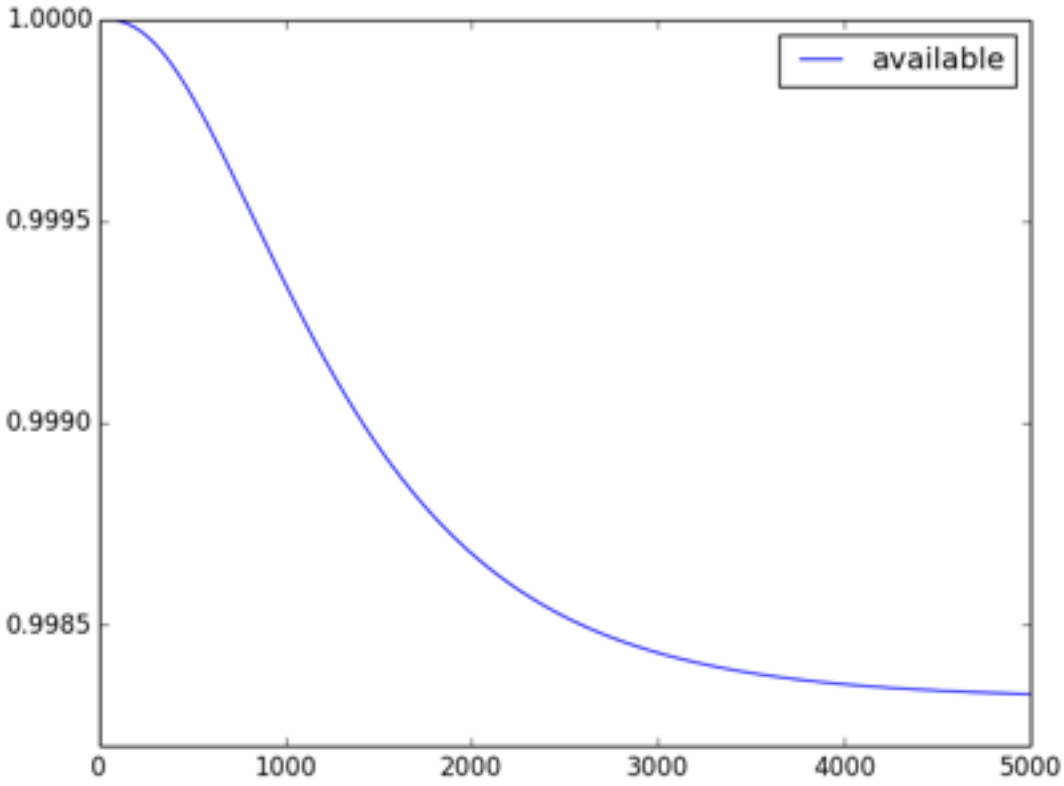


Courbes de disponibilité de notre application 2 en série

Valeur de la disponibilité asymptotique de l'application 2 en série :

```
print(process.value(5000, available))
#0.648582013752
```

La seconde version de notre application numéro 2 est un système où nos trois composants sont en redondance active :

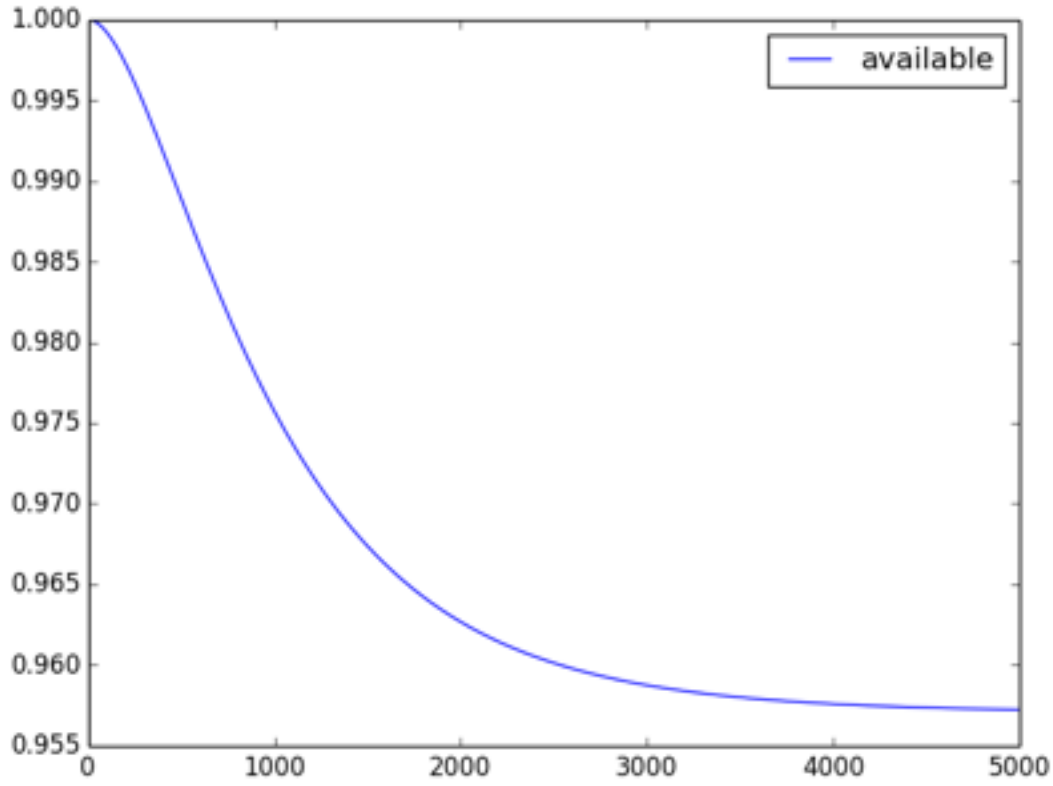


Courbes de disponibilité de notre application 2 en redondance active

Valeur de la disponibilité asymptotique de l'application 2 en redondance active :

```
print(process.value(5000, available))
#0.998327740731
```

La troisième version de notre application numéro 2 est un système où nos trois composants forment un voteur 2 parmi 3 :



Courbes de disponibilité de notre application 2 en voteur 2/3

Valeur de la disponibilité asymptotique de l'application 2 en voteur 2/3 :

```
print(process.value(5000, available))
#0.967231110811
```

Conclusion :

Ce TP nous a permis de nous familiariser avec l'utilisation de la technique des graphes de Markov sur fiabilipy dans le cas d'architectures simples à deux ou trois composants.

Nous avons pu rapidement nous rendre compte de la facilité que nous permet cette technique pour étudier les probabilités des différents états de nos systèmes. Même si cette technique nous paraissait plus compliquée que celle des arbres de défaillances *a priori,* cependant, après s'y être confronté pendant deux heures lors du TP nous sommes convaincus de son utilité et sa facilité de mise en place.

Annexes :

Vous trouverez en annexe de ce rapport le code source de nos quatre architectures différentes :

- *l'application1.py correspond à notre système de deux composants en redondance active.*

- *L'application2.py correspondant à notre système de trois composants en série.*

- *L'application3.py correspondant à notre système de trois composants en redondance active*

- *L'application3.py correspondant à notre système de trois composants en voteur 2/3*

Vous trouverez également les différentes courbes demandées pour chaque architecture ainsi que les captures d'écran nous ayant servi à rédiger ce rapport.

Enfin vous trouverez la solution que nous avons implémenté sous scilab pour répondre à la problématique de la partie 3 de ce TP.