



COMPTE-RENDU

TP numéro 2 [LO22]



11 AVRIL 2016

Marion Chan-Renous-Legoubin & Lavolette Etienne

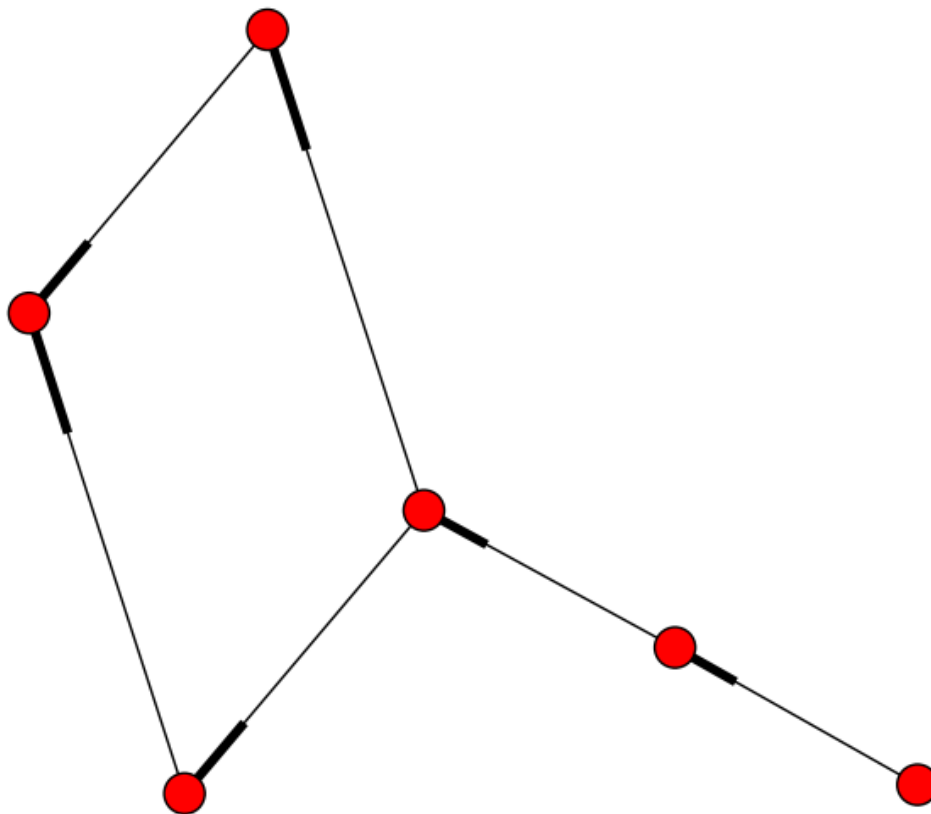
Rappel des objectifs du TP :

L'objectif principal de ce TP numéro 2 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 arbres de défaillance, en utilisant l'outil python "Fiabilipy". Ainsi, nous avons étudié quatre architectures différentes d'un système informatique. Nous avons déterminé l'arbre de défaillance pour chaque système grâce au logiciel "Arbre-Analyste", et modélisé les « blocs diagrammes de fiabilité » sur fiabilipy pour chacune des architectures (sauf la dernière car fiabilipy ne peut pas créer un système avec deux voteurs). Puis pour chacune d'entre-elles nous avons tracé les courbes donnant la disponibilité du système et calculer les facteurs d'importance de Birnbaum de chaque composant. Enfin nous avons proposé notre solution pour augmenter la fiabilité du système.

I. Architecture n°0

La première architecture à étudier est composée d'un bus de communication en série avec deux mémoires en parallèle (une seule est nécessaire pour le fonctionnement du système) en série avec trois processeurs : deux sont nécessaires pour que le système soit non défaillant : il s'agit donc d'un voteur 2/3. Nous avons modélisé ce système sur fiabilipy.

A l'aide de l'outil fiabilipy nous avons pu transcrire notre code en un bloc diagramme de fiabilité afin de comprendre l'agacement et le fonctionnement de notre réseau de stockage :

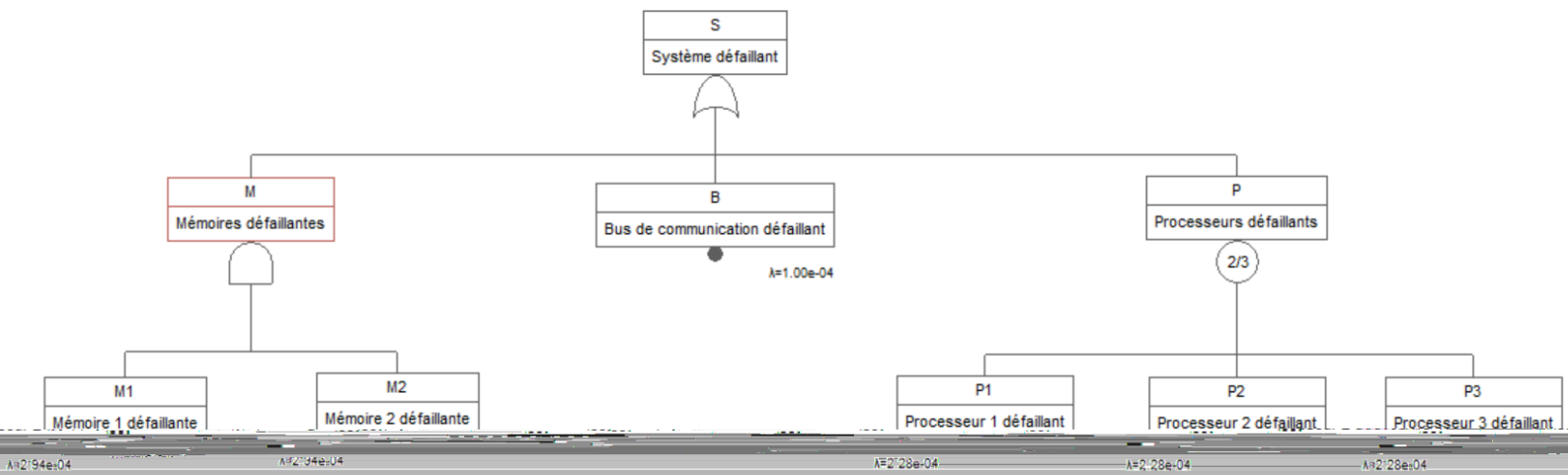


Bloc diagramme de fiabilité de l'architecture 0

Arbre de défaillance:

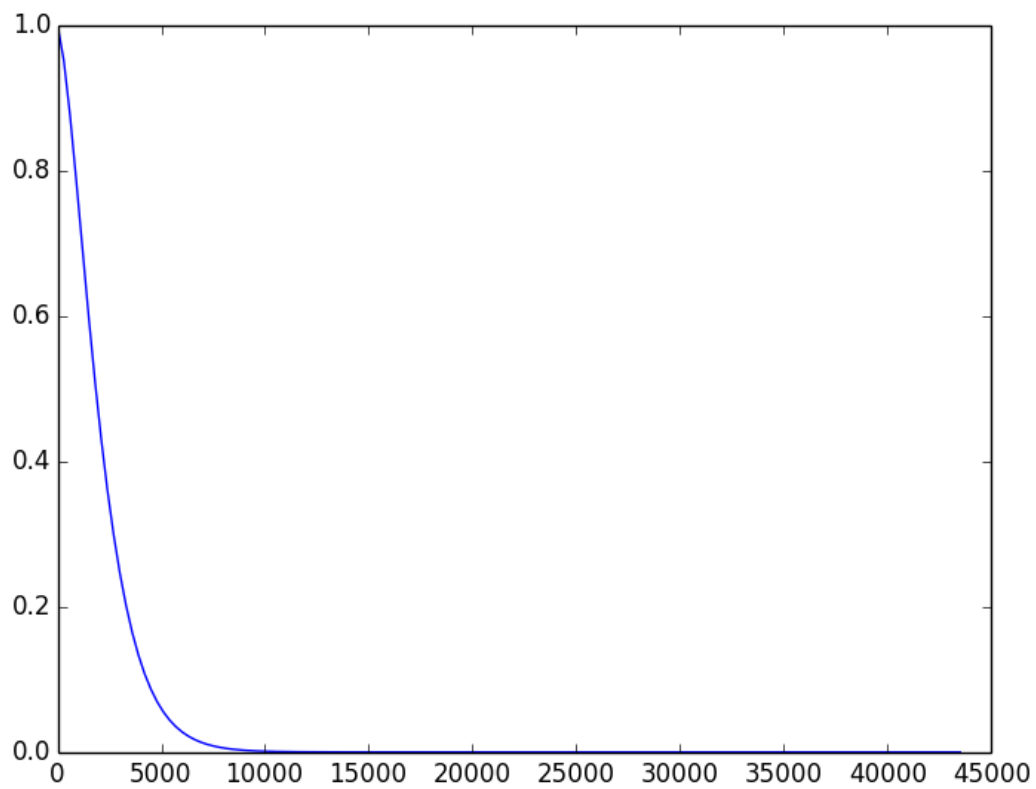
Compte rendu TP2

1.1 Grace au logiciel "Arbre-Analyste" nous avons pu déterminer l'arbre de défaillance de ce système. Les conventions du logiciel ne sont pas les mêmes que celles vues en cours: ici, les évènements élémentaires sont modélisés par des rectangles avec un cercle noir en-dessous, tandis que nous avons vu en cours qu'ils étaient modélisés par des cercles.



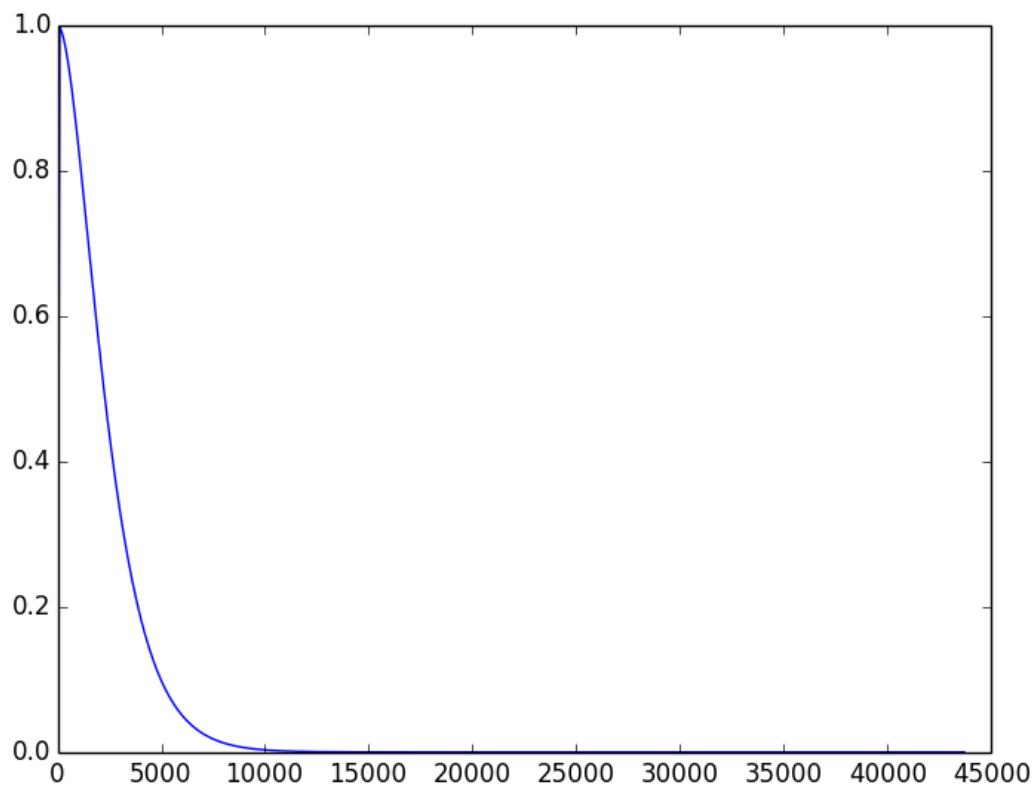
Arbre de défaillance de l'architecture 0

1.2 A l'aide du système modélisé en python, il nous a été très facile de tracer la courbe de disponibilité de notre première architecture à étudier :



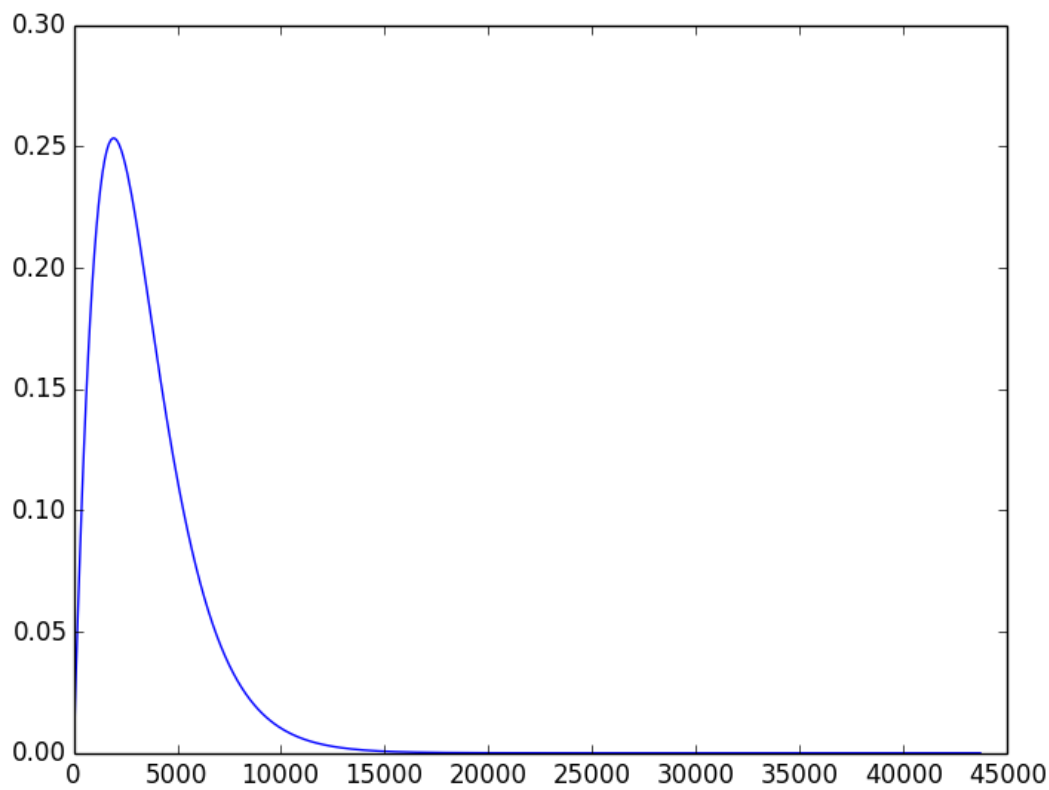
Courbe de fiabilité de l'architecture 0

1.3 Grâce à fiabilipy, nous avons pu calculer les facteurs d'importance de Birnbaum pour chaque composant de cette architecture.



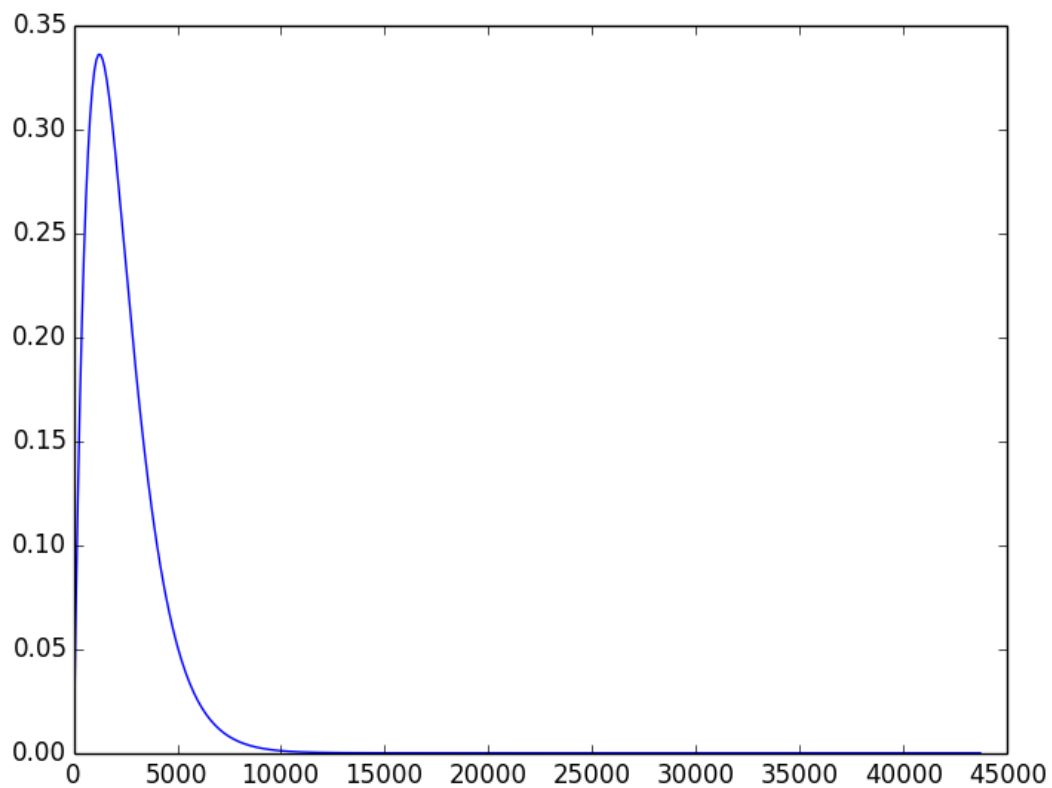
Facteur d'importance de Birnbaum par rapport au bus de l'architecture 0

Compte rendu TP2



Facteur d'importance de Birnbaum par rapport à une mémoire de l'architecture 0

Compte rendu TP2



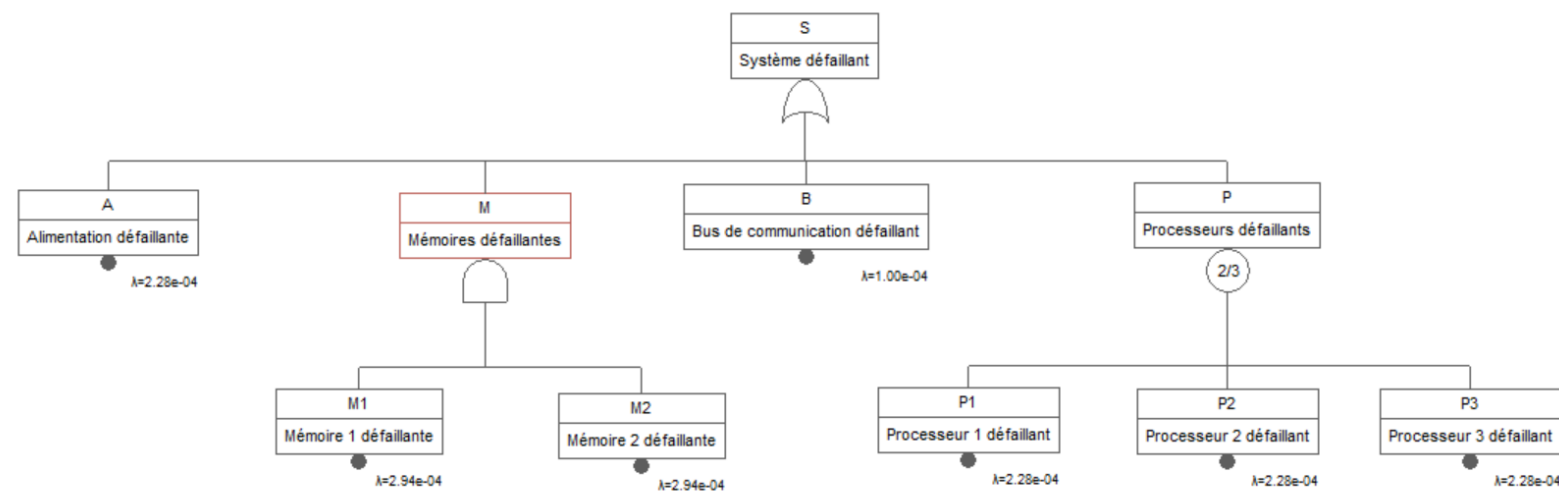
Facteur d'importance par rapport à un processeur de l'architecture 0

Conclusion : Au vu des facteurs de Birnbaum des différents composants de ce système, le bus est l'élément critique de ce système.

1.4 Afin d'améliorer la fiabilité de ce système, nous pourrions redondier le bus de communication et en mettre deux en parallèle.

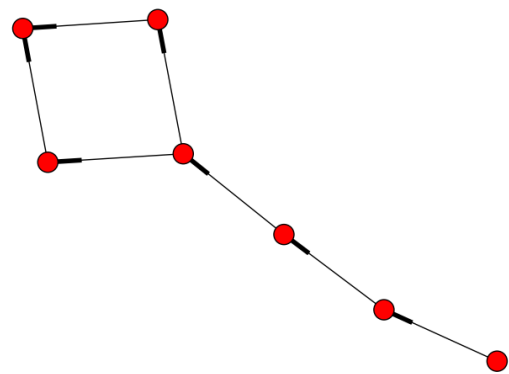
II. Architecture n°1

Dans cette seconde architecture nous ajoutons une source d'alimentation en série avec le système. Celle-ci doit fonctionner afin que le système informatique ne soit pas défaillant. Nous avons donc déterminé l'arbre de défaillance du système:



Arbre de défaillance de l'architecture 1

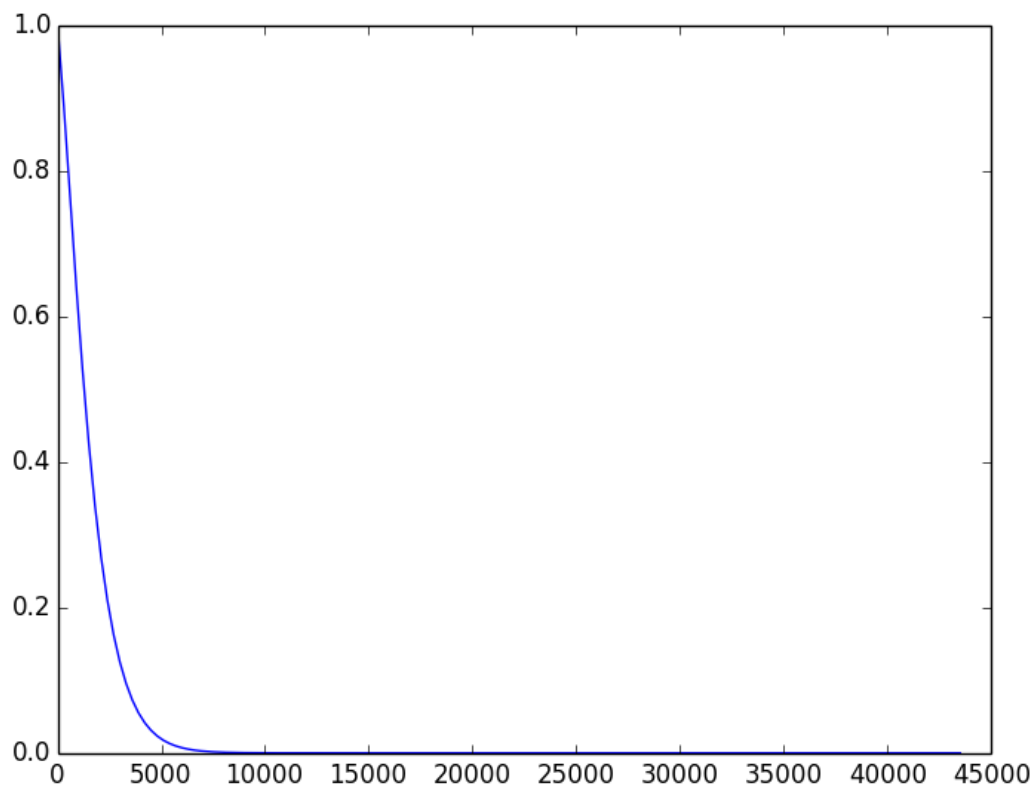
Voici le bloc diagramme de fiabilité de la seconde architecture.



Bloc digramme de fiabilité de l'architecture 2

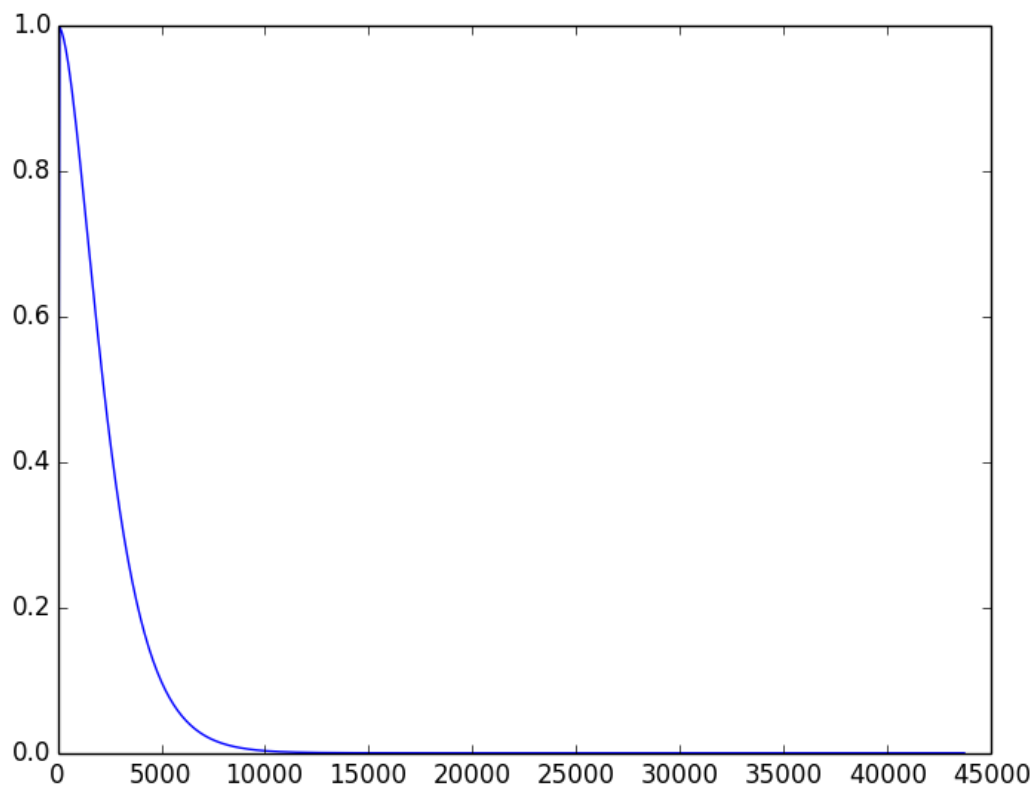
Nous avons tracé la courbe de disponibilité de cette architecture.

Compte rendu TP2



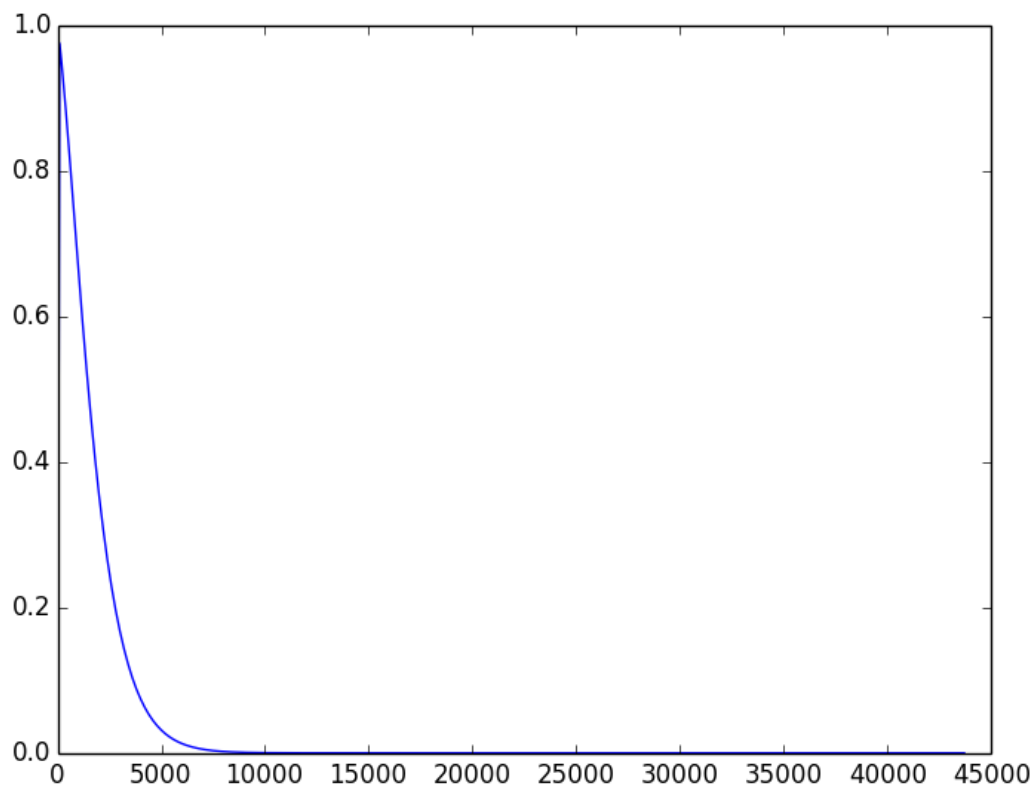
Courbe de disponibilité de l'architecture 2

Nous avons ensuite tracé les courbes correspondant aux facteurs d'importance de chaque composant du système.

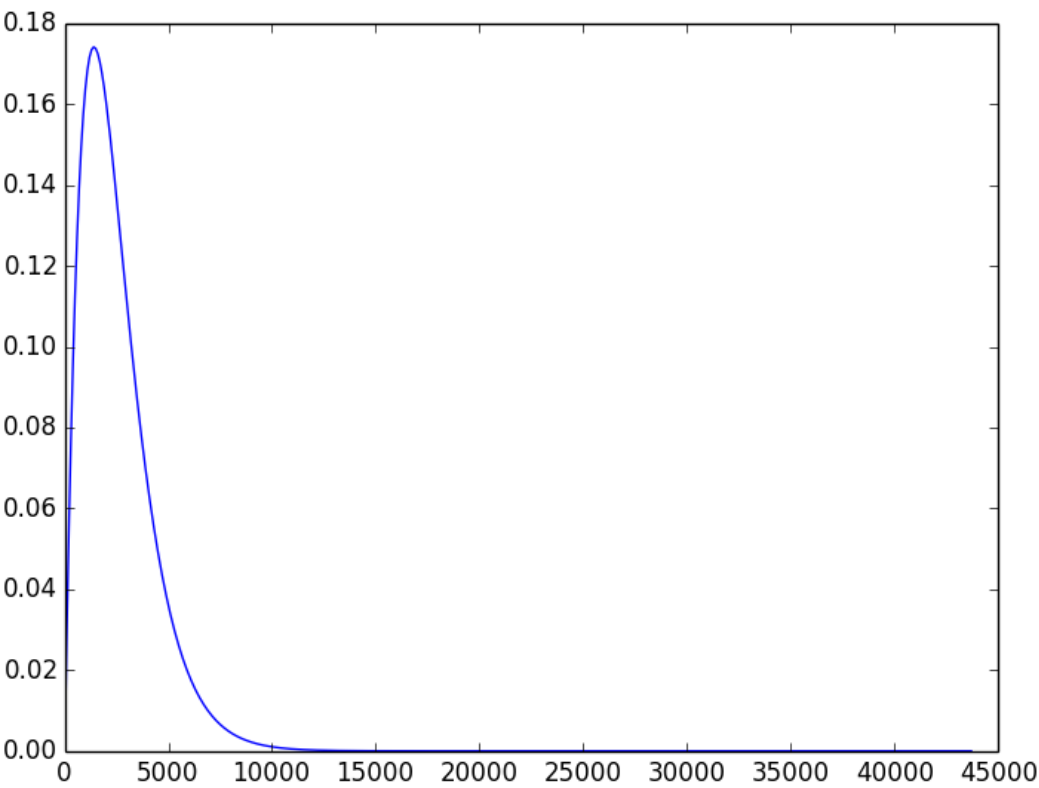


Facteur d'importance de Birnbaum de l'alimentation de l'architecture 1

Compte rendu TP2

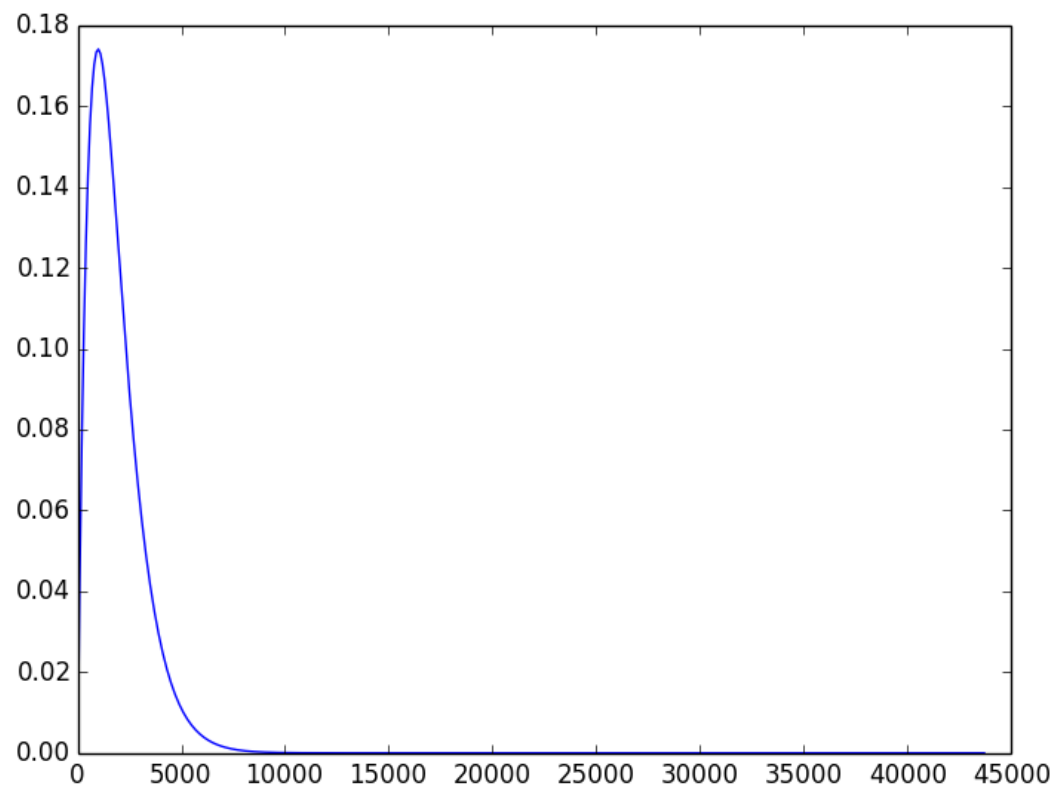


Facteur d'importance de Birnbaum du bus de l'architecture 1



Facteur d'importance de Birnbaum d'une mémoire de l'architecture 1

Compte rendu TP2

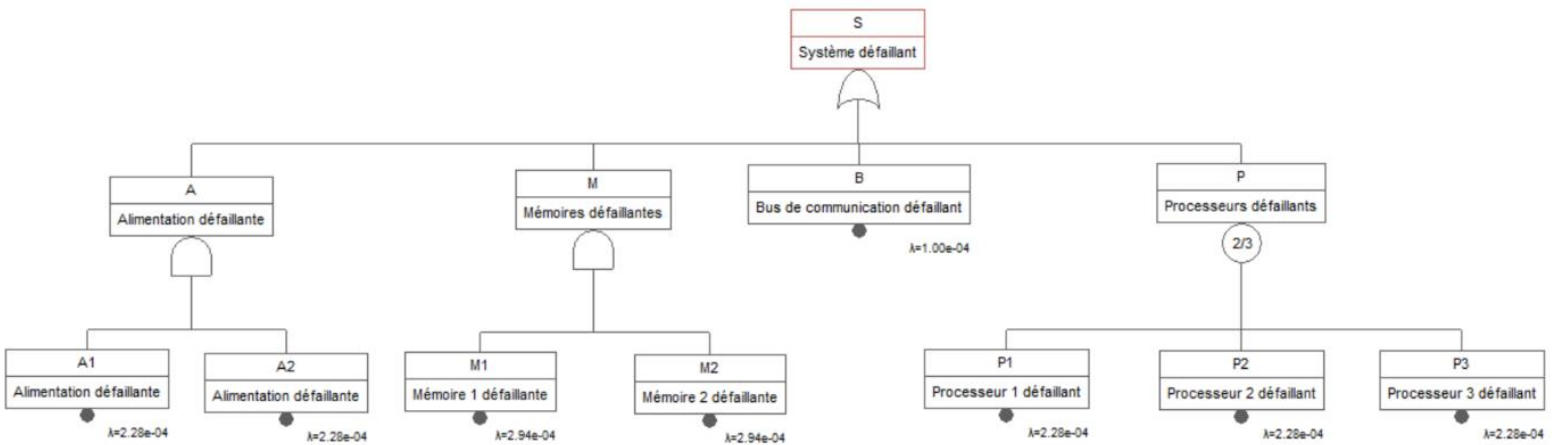


Facteur d'importance de Birnbaum par rapport au voteur de l'architecture 1

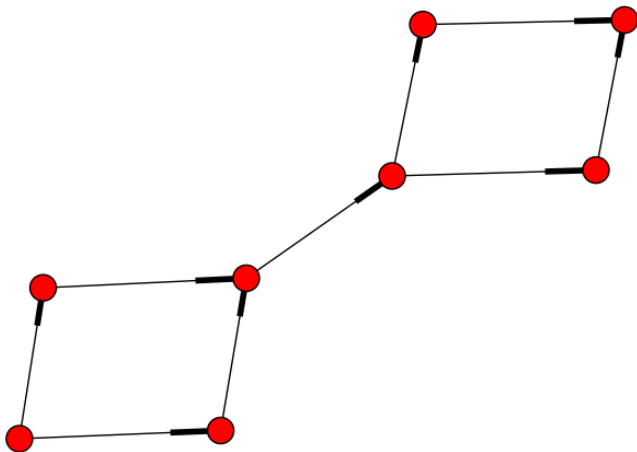
Conclusion : Nous pouvons voir ici que l'alimentation et le bus ont la même courbe et la courbe la plus élevée. Ainsi, ces deux composants sont les composants critiques du système.

III. Architecture n°2

Pour cette architecture, nous avons ajouté une seconde source d'alimentation, en parallèle avec la première source d'alimentation. Ainsi, afin que le système fonctionne, au minimum une des deux sources d'alimentation doit fonctionner. L'arbre de défaillance de ce système est le suivant :

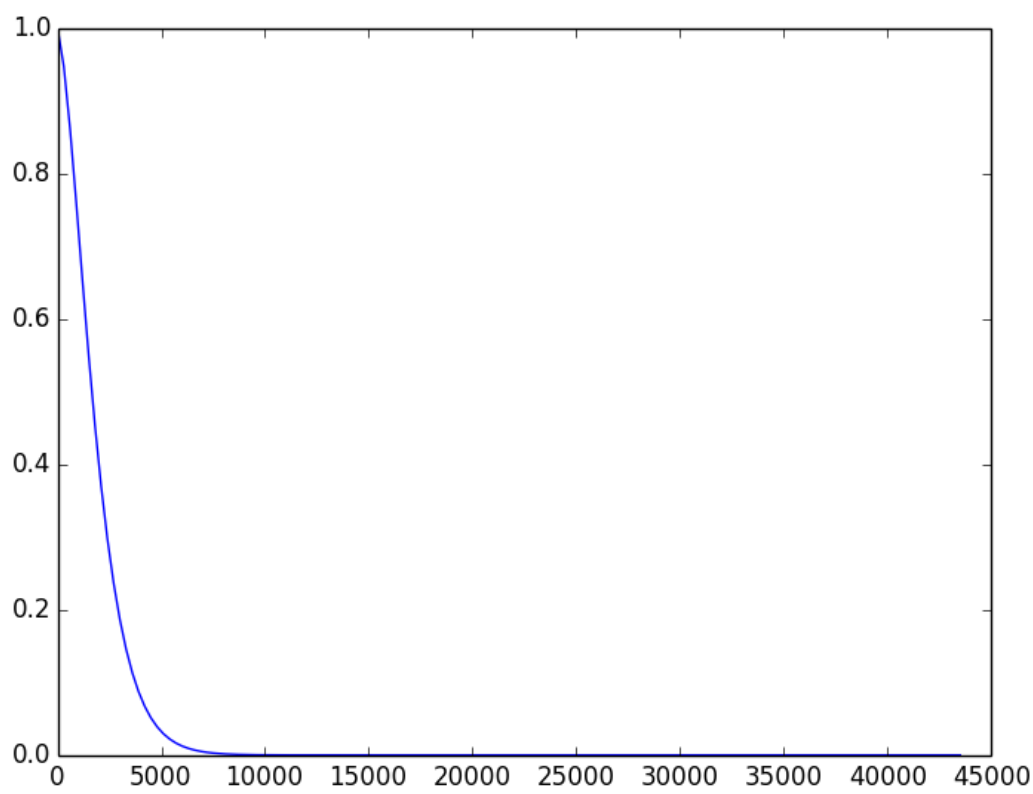


Arbre de défaillance de l'architecture n°2



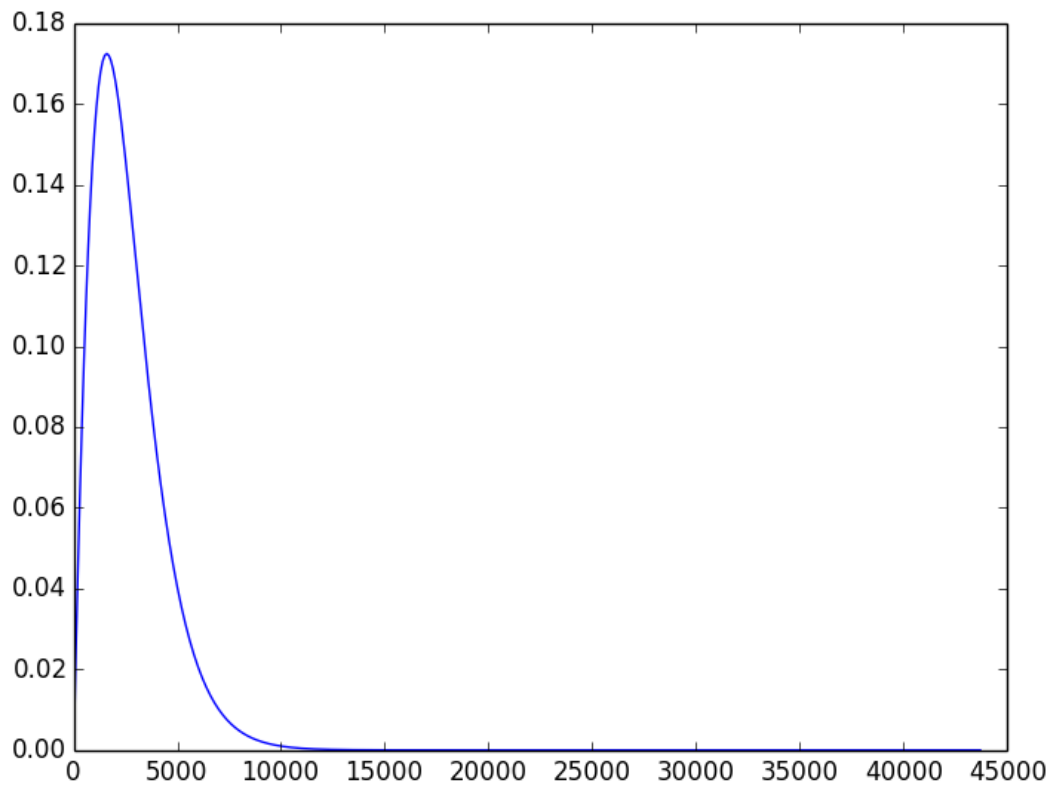
Bloc digramme de fiabilité de l'architecture n° 2

Voici la courbe de disponibilité de notre troisième architecture :

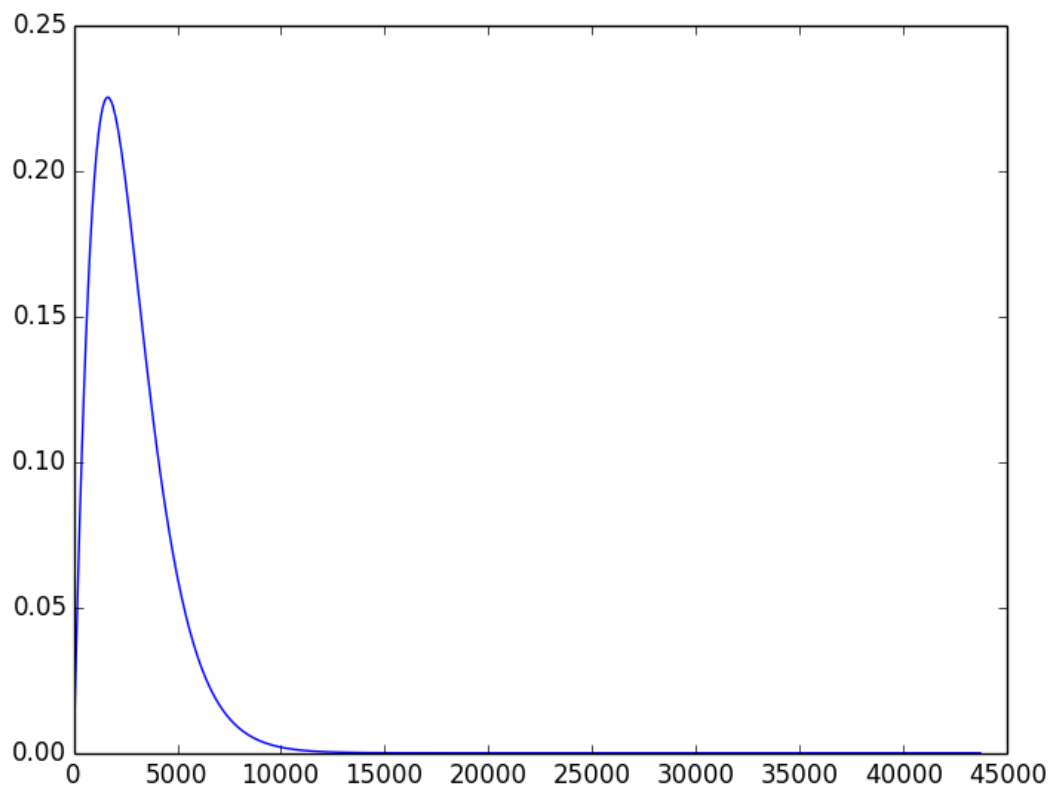


Courbe de disponibilité de l'architecture 2

Nous avons ensuite tracé les courbes correspondant aux facteurs d'importance pour chaque composant de ce système

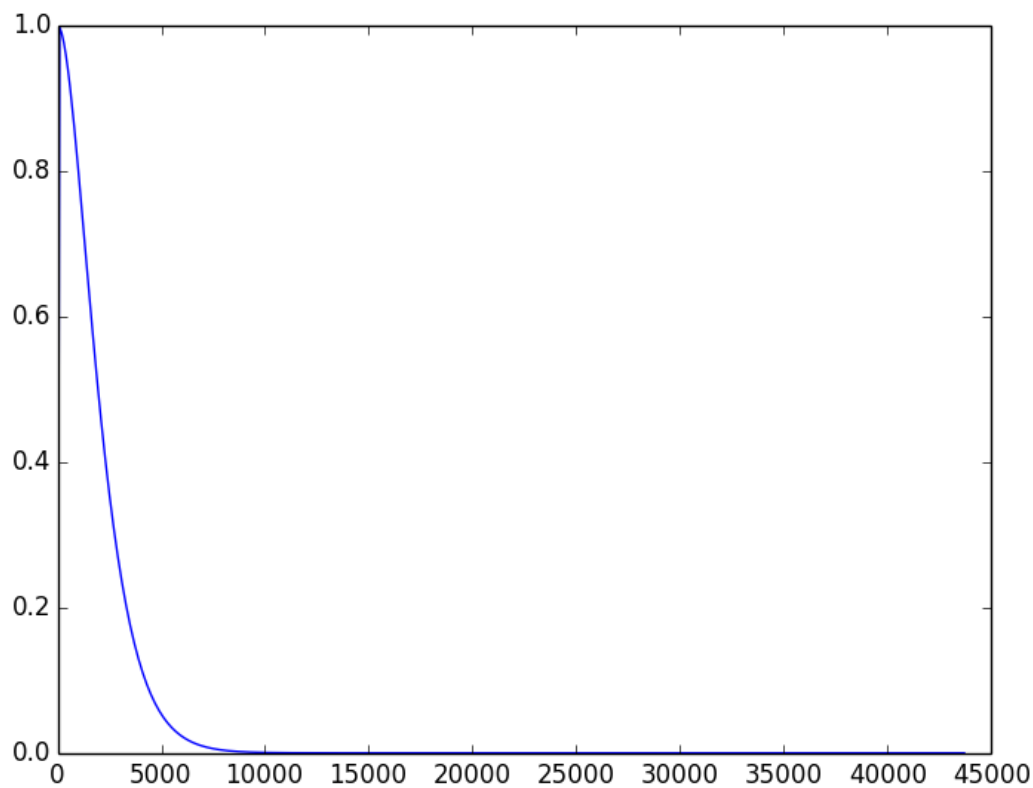


Facteur d'importance de Birnbaum d'une alimentation de l'architecture 2



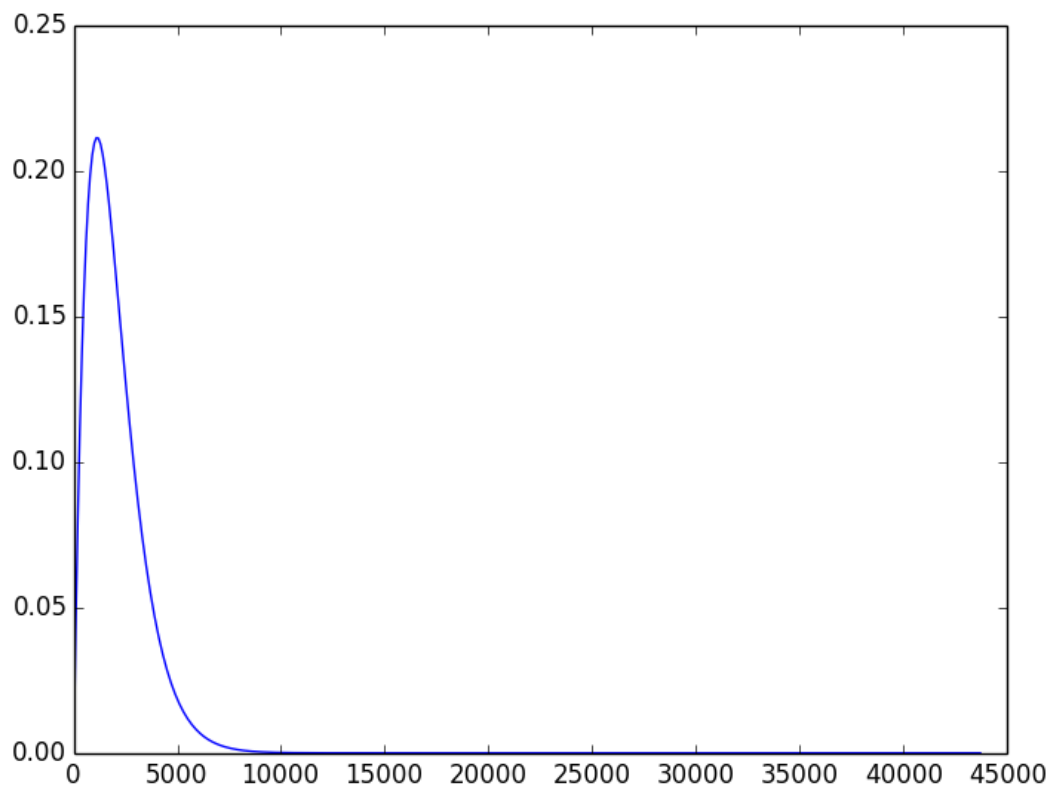
Facteur d'importance de Birnbaum d'une mémoire de l'architecture 2

Compte rendu TP2



Facteur d'importance de Birnbaum du bus de l'architecture 2

Compte rendu TP2



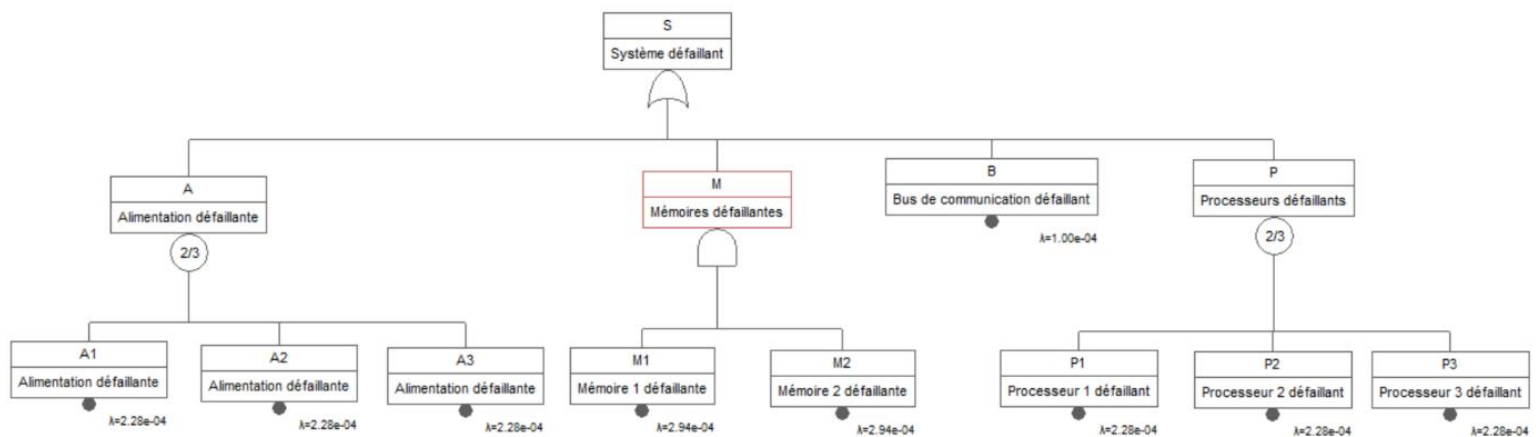
Facteur d'importance de Birnbaum du voteur de l'architecture 2

Conclusion : Nous pouvons voir que le voteur et le bus a la courbe la plus élevée : il s'agit du composant critique du système. Ainsi, on peut remarquer que l'alimentation n'est plus critique par rapport à l'architecture précédente, car nous avons redondé l'alimentation.

IV. Architecture n°3

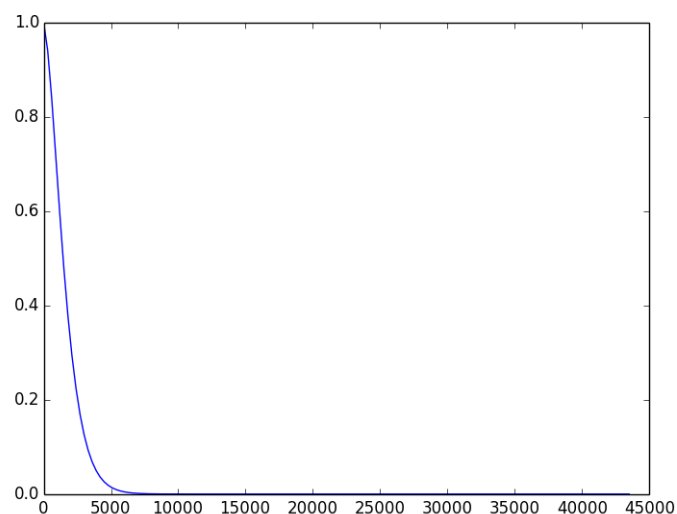
Pour cette dernière architecture, nous avons rajouté une troisième source d'alimentation au système, ce qui compose un voteur 2/3 avec les deux autres sources d'alimentation du système. Ainsi, pour que le système informatique soit non défaillant, il faut qu'au moins deux sources d'alimentation sur trois fonctionnent.

Nous avons déterminé l'arbre de défaillance de cette architecture :



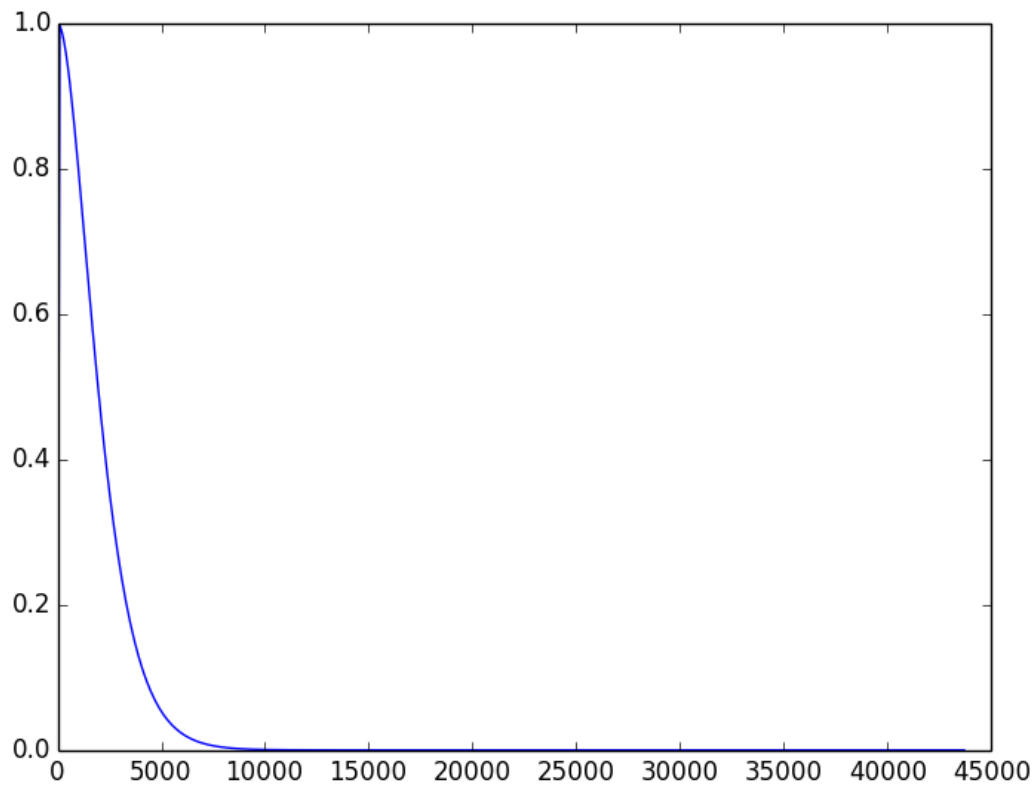
Arbre de défaillance de l'architecture n°3

Fiabilipy n'est pas en mesure de nous donner le bloc diagramme de fiabilité de ce système, mais nous pouvons cependant le modéliser sur le logiciel et tracer la courbe de disponibilité.

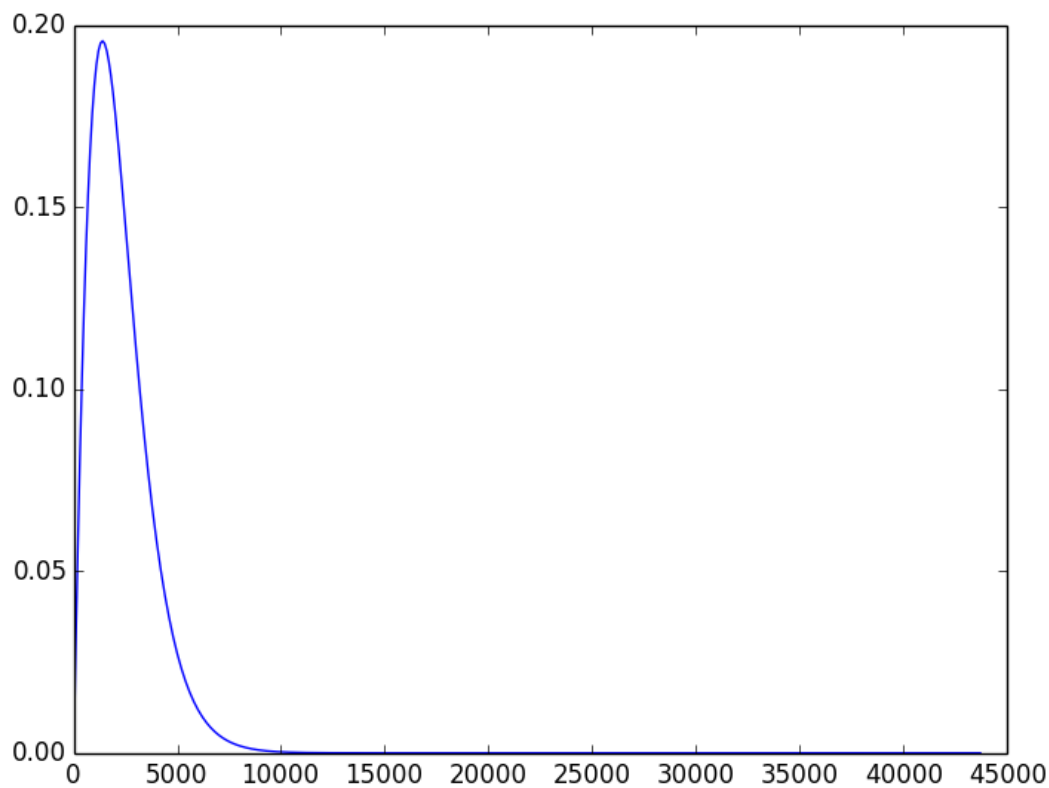


Courbe de disponibilité de l'architecture 3

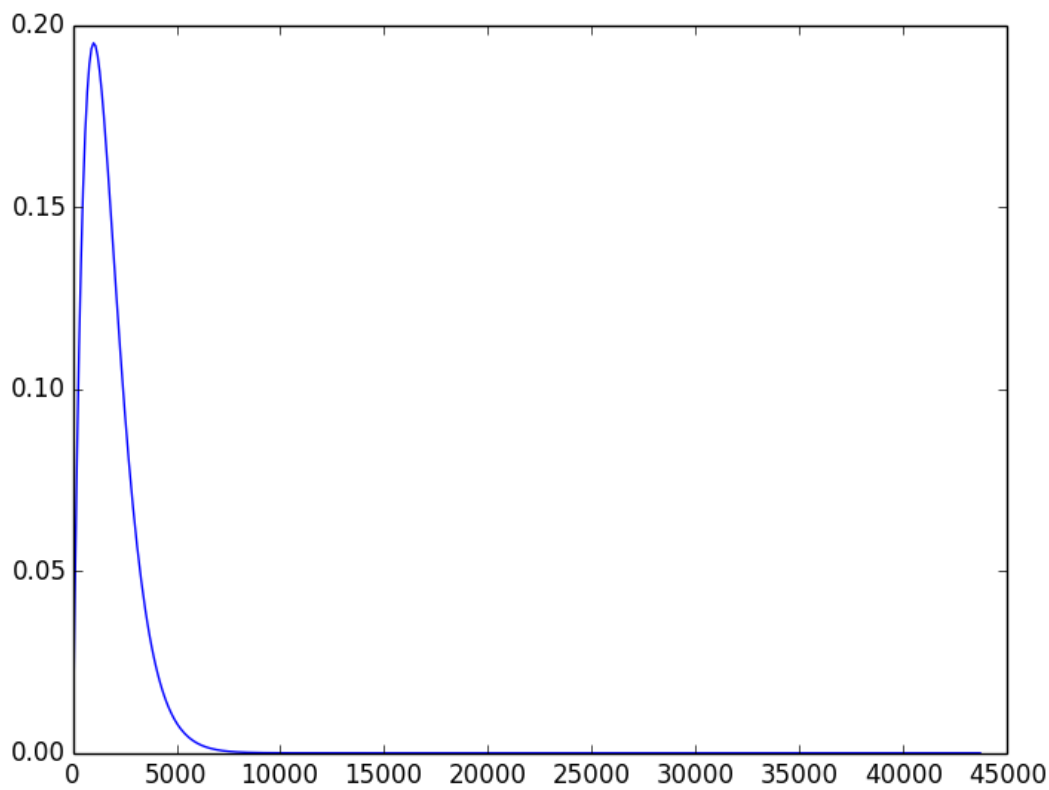
Nous avons ensuite tracé les courbes correspondant aux facteurs d'importance de Birnbaum pour chaque composant de cette dernière architecture.



Facteur d'importance de Birnbaum du bus de l'architecture 3

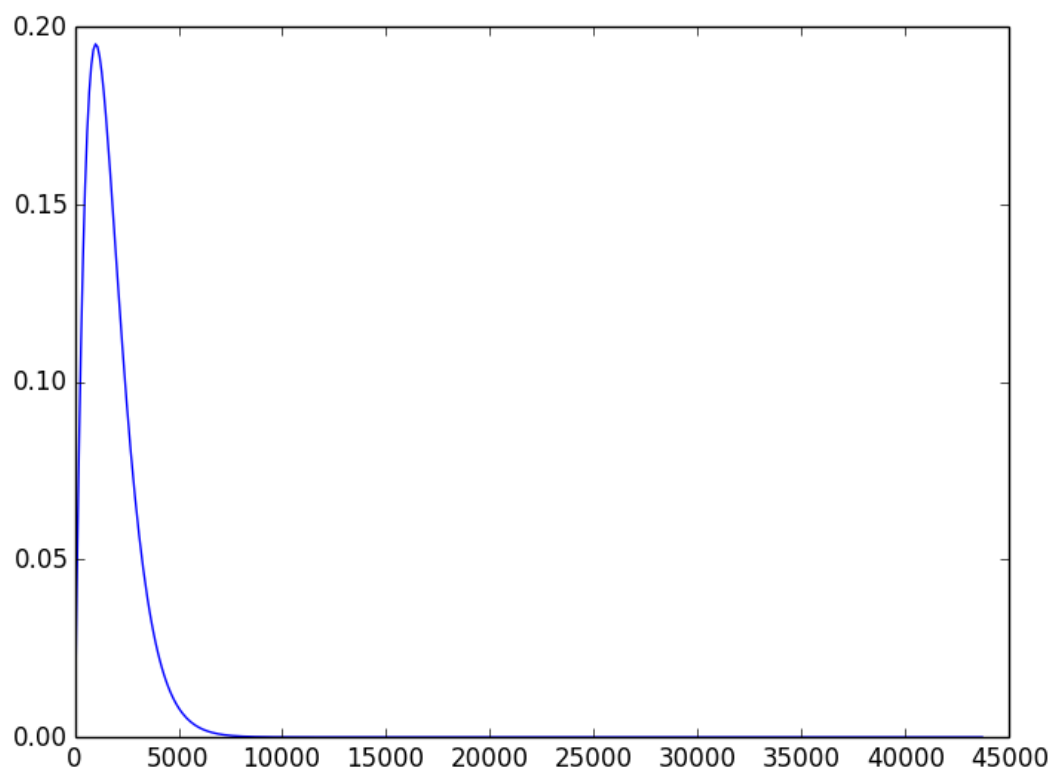


Facteur d'importance de Birnbaum d'une mémoire de l'architecture 3



Facteur d'importance de Birnbaum du voteur de processeur de l'architecture 3

Compte rendu TP2



Facteur d'importance de Birnbaum du voteur d'alimentation de l'architecture 3

Conclusion : Nous pouvons voir qu'à nouveau et le bus est l'élément critique du système.

Compte rendu TP2

Annexes :

Vous trouverez en annexe de ce rapport les code source des 4 architectures, ainsi que les courbes obtenues par fiabilipy pour l'exploitation des résultats. Pour chaque architecture, sont présents :

- *le code source simple*
- *le Birnbaum généralisée (code inexploitable mais donnant la technique utilisée pour calculer le facteur d'importance de chaque composant)*
- *des exemples de code pour calculer le facteur d'importance de composants particuliers choisis en exemple.*