

## TP 4 Partie 1 — Calcul de $\pi$ par une Méthode de Monte Carlo.

Implémentation pour environnement à mémoire partagée.

Dans cette première partie de TP nous souhaitons développer un code Monte Carlo (MC) pour le calcul de  $\pi$  sur une architecture à mémoire partagée.

Nous allons au cours de deux séances :

- Analyser l'algorithme MC et proposer deux variantes possibles suivant un modèle de programmation par tâche :
  - Parallélisme de boucle, une tâche par itération
  - Structure Maître Esclave, plusieurs itérations par tâche
- Etudier deux implémentations en Java pour architecture à mémoire partagée :
  - Code 1 : Assignment 102, Auteur : Karthik Jain (<https://www.krthkj.com>)  
<https://gist.github.com/krthkj/9c1868c1f69142c2952683ea91ca2a37>
  - Code 2 : Pi.java, Dr. Steve Kautz, IOWA State University (<https://faculty.sites.iastate.edu/smkautz/>)  
[http://web.cs.iastate.edu/~smkautz/cs430s14/examples/thread\\_pool\\_examples/pi/Pi.java](http://web.cs.iastate.edu/~smkautz/cs430s14/examples/thread_pool_examples/pi/Pi.java)

Ensemble nous étudions l'algorithme MC que vous présenterez dans votre rapport. Puis nous proposons une ou plusieurs parallélisations de l'algorithme. Enfin nous étudions les deux implémentations proposées et nous les analysons.

Le rapport qui couvrira ce TP ainsi que sa suite comportera : une introduction et une conclusion, ainsi que trois parties suivant le plan de TP proposé. Une quatrième partie viendra compléter ce rapport afin de traiter de l'implémentation sur machine à mémoire distribuée. En fonction de l'avancement, une cinquième partie pour être ajoutée afin de traiter de l'implémentation sur deux niveaux de parallélisme.

### Exercice 1. : étude de la méthode de MC

Présentez la méthode de Monte Carlo et proposez un algorithme séquentiel.

### Exercice 2. : Parallélisation

Proposez une ou plusieurs parallélisations possibles de l'algorithme MC.

Pour chaque algorithme proposé on présente le modèle de programmation parallèle suivi, le ou les paradigmes choisis, ainsi qu'une analyse de la complexité des algorithmes.

### Exercice 3. : Mise en oeuvre

Etudiez deux implémentations en Java pour architecture à mémoire partagée :

- Code 1 : Assignment 102, Auteur : Karthik Jain (<https://www.krthkj.com>)  
<https://gist.github.com/krthkj/9c1868c1f69142c2952683ea91ca2a37>
- Code 2 : Pi.java, Dr. Steve Kautz, IOWA State University (<https://faculty.sites.iastate.edu/smkautz/>)  
[http://web.cs.iastate.edu/~smkautz/cs430s14/examples/thread\\_pool\\_examples/pi/Pi.java](http://web.cs.iastate.edu/~smkautz/cs430s14/examples/thread_pool_examples/pi/Pi.java)

Pour chaque code :

- quel est le modèle de programmation parallèle suivi ainsi que le paradigme choisi ?
- Est-ce que le code correspond à ce qui a été proposé en exercice 2 ?
- Analysez la scalabilité forte et faible de chaque code.
- Quelle implémentation conseillez vous ?