# PROJET DE RESOLUTION DE SUDOKU PAR LE SOLVEUR SMT AVEC Z3

REDIGÉ PAR :

* Daly SISSOKO
* KOFFI N’GUESSAN Joel
* Fitzgerald NGUENA

## Objectifs du travail

L’objectif du projet est de développer notre propre solver de Sudoku distribués dans Spark par le silver SMT Z3 et de vérifier à l’exécution les gains de parallélisation sur un cluster à plusieurs nœuds et sur un cluster à un seul nœud.

## DEMARCHES ET RESOLUTION DISTRIBUÉE DU SUDOKU

A l’aide du Sudoku Benchmark, nous avons commencé par benchmarker tous les solvers SMT Z3 disponibles dans le Z3 Solver.cs observable dans l’explorateur de solutions. Pour ses bonnes performances en termes de temps de résolution, nous avons choisi le solver Z3AssumptionsSolver. Le programme Z3AssumptionsSolverSpark est intégré dans le répertoire de sortie. Nous avons ajouté le code du solver au « main » sous la forme d’une classe qui va résoudre une grille de Sudoku et nous avons ajouté le nuget Microsoft.Z3 et Microsoft.Spark pour gérer les instances.

Nous avons créé une fonction appelée « solveSudoku » qui prend en entrée une chaîne de caractère et retourne un résultat de type chaîne de caractère ; de manière pratique, la fonction prend en entrée une grille de Sudoku et renvoie comme résultat une chaîne caractères. A cela, nous avons implémenté le programme « Static void main » qui prend comme argument une grille de Sudoku avec un switch qui permet de choisir l’exécution soit avec un nœud soit avec cluster de plusieurs nœuds.

On a créé un dataframe avec spark.read qui permet de lire et de transformer les colonnes « quizzes string » et « solutions string » en un dataframe. Ce dataframe contient l’ensembles des Sudoku qu’on va réduire en un dataframe de faible taille par l’instruction LimitedSudoku et une UserDefineFunction.

## RÉSULTATS

Au final, nous avons réussi à implémenter la résolution distribuée des Sudoku sous Spark en utilisant le solver SMT Z3. Nous constatons aussi que les gains de temps de résolution un nœud vs un cluster de nœud n’étaient pas très significatifs en dessous de 10000 sudoku à résoudre. Au-dessus de ce nombre nous observons que le temps résolution avec un cluster de nœuds est beaucoup plus faible que celui avec un seul nœud. Nous avons aussi observé que le temps de résolution est très long à partir de 500000 sudokus (20 min d’attente sans résolution). Nous considérons que notre programme Spark ne peut aller au-delà d’une résolution de 500000 sudoku.