TP BigData et performance de l'analyse de configuration Intense - ☺

- STEP 0 : Ecrit un script qui copie le fichier « router.unix » 10.000 fois dans un répertoire « conf » local sous la forme « router.unix.x »

  Prendre le nom de « generate.sh » pour utiliser le Makefile : « gmake generate » pour l'appel

- STEP 1 : Ecrire un programme awk optimal (utilise la commande next) qui détecte si « ip access-group » est mis sur les interfaces du routeur (avec contrôle de bloc !!). La sortie du programme sur router.unix doit donner exactement cette sortie:

  ```
  ./router.unix interface FastEthernet0 missing ip access-group
  ./router.unix interface GigabitEthernet0/1 missing ip access-group
  ```

  Prendre le nom de « check.int.sh » pour utiliser le Makefile.run : « gmake -f Makefile.run step1 » pour l'appel

  AJUSTER le nombre de fichiers (STEP 0) pour que le temps de traitement de la STEP 1 soit de l'ordre de 2 minutes sur votre système.

  Vérifier que le fichier de sortie est consistent : on appliquera ce petit awk ☺

  ```
  awk '{
          print "field2_length : "length($2);
          print "field3_length : "length($3);
          print "field4_length : "length($4);
          print "nb_of_fields : "NF;
  } END {
          print "number_of_lines : "NR;
  }' $1 | sort | uniq -c | column -t
  ```

  Notes :

  - Chercher des informations sur « xargs »

- STEP 2 (rien à rendre) : Lance « gmake -f Makefile.run step2 ». Est-ce plus rapide et peut-on assurer que le fichier de sortie est consistant ?

  Notes : le fichier de sortie est consistent ?

- STEP 3 : Ecrire un programme flex qui détecte si ip access-group est mis sur les interfaces du routeur (avec contrôle de bloc !!). Lance le sur toutes les confs du répertoire « conf » et note le temps d'exécution (« time pgm »).

  Prendre le nom de « check.flex.l » pour compiler avec « gmake check.flex » et « gmake -f Makefile.run step3 » pour l'appel

  Notes : le fichier de sortie est consistent ?

- STEP 4 (rien à rendre) : Lance « gmake -f Makefile.run step4 », est-ce plus rapide et le fichier de sortie est-il consistant ?

  « gmake -f Makefile.run step4 » pour l'appel

- STEP 5 : Modifier le pgm check_flex.l en check_flex_race.l pour ajouter une écriture dans un fichier unique par processus (race conditions) (redirige la sortie vers le fichier zz.output.step5) :

  a. File *output

  b. Générer un fichier unique dans le répertoire local « tmp » en utilisant « getpid() » dans le nom du fichier temporaire.

  ```
  $ ls tmp

  10000   10056   10112   10168   10224   10280   10336   10392
  10448   10504   10560   10616   10672   10728   10784   10840
  10896   10952   11008    9464    9520    9576    9632    9688
  9744    9800    9856    9912    9968

  $
  ```

  c. Modifier printf("..") en fprint(output, "..")

Récupère tous les fichiers temporaires générés dans « tmp » et met les dans « zz.output.step5.txt », a-t-on l'assurance que le fichier de sortie sera alors consistant ?

Prendre le nom de « check.race.l » pour compiler avec « gmake check.race » et « gmake -f Makefile.run step5 » pour l'appel.

- STEP 6 : Ecrire un programme FLEX qui compte le nombre de mot « ip » et « any » dans le fichier router.unix.

- STEP 7 : Ecrire un programme FLEX qui vérifie que access-class in && out sont mis sur toutes les lignes vty dans le fichier router.unix.

- STEP 8 : Ecrire un programme FLEX qui vérifie que la consistance des ACLs defs pas refs et refs pas defs dans le router.unix. On pourra reprendre le programme en awk et contrôler les performances avec le programme en FLEX sur les milliers d'instances crées de router.unix dans le cadre du TP.

on commence à détester le prof

- STEP 9 : Ecrire un programme C ou autre langage qui dit si une chaine de caractère en entrée mord ou pas sur la regex « (a|b)ab » **en implémentant un FA (ref doc automate.pdf).**

il commence à pousser le bouchon un peu loin !!

- STEP 10 (exercice de style et super bonus !!) : Ecrire un programme FLEX qui dit si une chaine de caractère en entrée mord ou pas sur la regex « (a|b)ab » **en implémentant un FA via les états FLEX.**