*BAYO Samedi , 26 / 10 / 2019*

*Jordy Riguel*

*La Sentence*

*LIR – IUTFV*

**ADMINISTRATION SYSTEMES**

**TP 1 : Systeme de Monitoring Moniotring fait Python3**

**1 INTODUCTION**

Le Monitoring est un processus qui permet de verifier de facon permanente l’etat(de securite) d’une machine ou d’un Agent dans le reseau.

**2 Synthese du Projet**

L’enjeu du TP est de pouvoir realiser le monitoring de plusieurs capteur du systeme appele “sensors”. Le systeme est constitue d’un Serveur (host) et de plusieurs clients appele ou capteurs(sensors) identifier par un unique identifiant qui est son adresse IP(Internet Protocol), qui accedent tous a une meme ressource web http://%s:%s/xmlrpc/" % (host, str(port)

deployer par un serveur cherrypy.

L ‘objectif ultime etant de pouvoir verifier l’etat de sante de chaque sensor sur le reseau et ensuite d’envoyer une alert(porter une exception) au serveur si le Monitor\_daemon rencontre un soucis.

Chaque etat de systeme, configuration, tache a operer(probescheduler et monitor scheduler), insertion doit etre ensuite sauveguarder dans un base de donnees sqlite3.

**3 Analyse technique et Implementations**

Au niveau de l’implementation technique s’est effectue en plusieurs etapes a savoir ,

- la modelisation suivi de la creation de la base de donnee

- support python(mettre en place le code xmlRPC et cherryPy) pour pouvoir faire communique les elements du reseau et deployer le serveur web

- Les process serveur consistent tout simplement a pouvoir sauveguarder toute les

nouvelles configuration systemes(ie nombre de sensor par exemple, ip etc…)

- Code python permetant de donner les authorisation au serveur de lire les ressources system(probeE\_Reading)

- soumettre les nouvelles configuration issue de la Base de Donne (SELECT Hostparams)

- Envoie de nouveau code (self.cm.sensor.source\_dir) pour lire les ressource

– checking de la sante du serveur(Host)

- Scheduler toutes les taches qui doivent etre lancer grace au probescheduler

- Lancer Multiple process grace au multithreading et multiprocessing

- implementation des classes d’extension (clocktimer, ticket\_scheduler, )

- lancer les methodes a des intervalle egale de temps

- utilisation des classes

**4 Implementations**

le TP a ete implementer em plusieurs fichier

\* currentTime.py (prise du temps reel )

\* monitor\_cli.py (execution des commandes cote client avec xmrpclib)

\* monitor\_daemon.py (lancement du programme Root(\) xml)

\* monitorLib.py (fichier contenant les classes scheduler et ticketscheduler pour lire la BD sqlite)

\* monitor\_db\_init.db (fichier sqlite)

**5 Problemes rencontres**

**Le veritable problemes rencontre durant ce projet est le versionning des bibliotheques python de 2.0 a 3.7.3 , Pour palier a cela nous avons creer un environnement python 2.7 qui est l’une des derniere version de python disponible encore en ligne mais dont les packages de certaines bibliotheque n’exhistent plus ou ont ete update, du coup certaines methodes tels que (from cherrypy import \_cptools, from ConfigParser import SafeConfigParser). Afin de mieux comprendre cette notion au mieux nous avons juger necessaire de refaire le TP de facon plus ameliorer avec python3 et ses bibliotheques disponible.**

**TP 2 : Systeme de Monitoring Moniotring fait Python3 d’un client via socket.scoket et putils(version moderne de \_cptools)**

Pour le faire nous avons creer deux dossier (Agent et Host).

- Le Dossier host contient les fichier dedie au serveur(config) a savoir sever.py permetant d’initialiser la communication entre le serveur et les agents

- Le Dossier Agent contient les fichiers(client.py, probeReading.py, sendError\_byEmail.py)

\* client.py permet de donner les authorisations et initialisation de la communication

\* probeReading.py qui permet de lire les ressources systems (cpu\_user, cpu\_system, cpu\_freq, memoire\_ram, sotrage\_disk)

\*sendError\_byEmail utilise les bibliotheques urlib3, smtplib de python3 pour envoyer une alerte a l’administrateur en cas de overlimit ou de surchauffe du processe ou du non respect des conditions definits.