

Gestion de projet ETRS001P

Table des matières

I.	Introduction	2
II.	Direction choisie	2
a.	Module de configuration	3
i.	Introduction	3
b.	Module de surveillance	4
c.	Module de Log	5
III.	Planning	5
IV.	Digrammes	6
a)	Diagramme de cas d'utilisation	6
b)	Diagramme de la base de donnée SQLite	6
V.	Les OID	6
a)	Requête GET	7
b)	Requête snmp Walk	8
VI.	Difficultés rencontrées	8
VII.	Amélioration future	9
VIII.	Conclusion	9

I. Introduction

Le projet est de développer une application (outil) de supervision de matériel réseau via SNMP. Le but est de pouvoir :

- Surveiller en ligne l'ensemble du matériel constituant le parc. Cette surveillance pourra avoir une représentation textuelle et graphique
- Sauvegarder (en fichier ou base de données) l'ensemble des données issues du matériel
- Gérer les défaillances matérielles (message d'erreur) en les reportant sur les interfaces « utilisateur » ainsi que dans le système de « log »

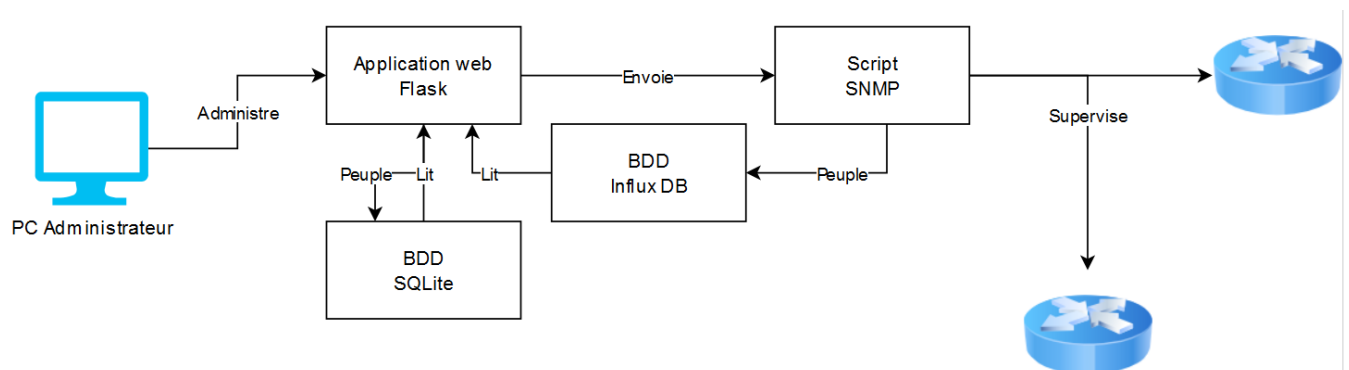
Pour cela l'application a été divisée en 3 gros modules :

- Le module de configuration : ce module permet d'ajouter, modifier et supprimer du matériel en fonction du besoin de l'utilisateur. Entre autres il est possible de préciser des informations supplémentaires telles que le type de matériel, l'élément à surveiller, les informations à récupérer,...
- Le module de surveillance : Ce module applique la surveillance indiquée sur les équipements et les éléments indiqués dans le module de configuration. Cette surveillance est effectuée en SNMP via une interrogation périodique. Les éléments surveillés sont ensuite affichés (texte et graphique)
- Le module de log : Ce module permet de faire une trace de tous les éléments des équipements surveillés. Chaque information récupérée est taguée et enregistrée dans un fichier pour permettre une consultation et un archivage de l'historique.

II. Direction choisie

Dans cette partie, nous allons vous expliquer les différents choix et notre vision pour le développement ainsi que pour le bon déroulement du projet.

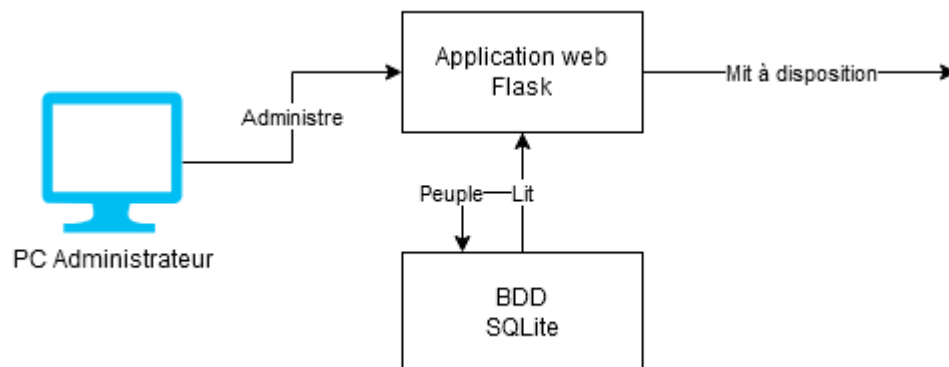
Notre application générale sera sous la forme d'un client léger pour faciliter son utilisation ainsi que son déploiement. Ce choix se traduit par l'utilisation de module tout aussi léger pour les différentes parties telles que du Flask pour le serveur web, etc... Ci-dessous un aperçu général de notre application.



a. Module de configuration

i. Introduction

Le module de configuration est le premier module sur lequel nous avons travaillé. Ce module est la base de notre projet car il permet de peupler une base de données qui sera consultée par les modules suivants. Ci-dessous la partie de notre module configuration



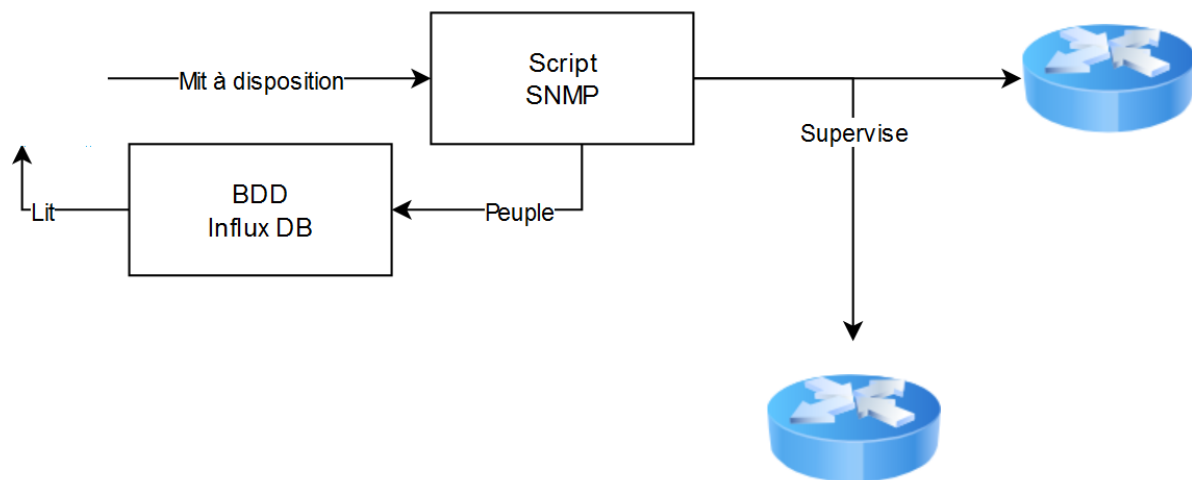
Pour que l'administrateur puisse indiquer ce qu'il veut surveiller, nous avons fait un serveur web en Python avec le framework Flask. Ce choix a été fait dans le but d'avoir une application légère et très simple. En effet le framework flask est extrêmement léger et puissant. De plus, l'ayant déjà manipulé l'année dernière dans le cadre d'un projet de M1, nous le connaissons et le maîtrisons bien.

Cette application web est en lien avec une base de donnée. L'application peuple et lit la base de données. Cette base de données est faite en SQLite. De même que pour Flask, le choix d'une base de données en SQLite est tout simplement pour sa légèreté et aussi parce que nous l'avons déjà manipulé. Cette base de données stocke toutes les infos fournies via le module de configuration. Ce qui veut dire que des infos telles que le nom de la machine, l'adresse IP,... seront présents dans cette base de donnée. Cette base est aussi dynamique pour l'ajout, suppression et modification des données des machines.

Le module de configuration (Application web et base de données) communique avec le module de surveillance. L'application Flask met à disposition les différentes données de la BDD SQLite. Ces données sont fournies au format JSON pour une meilleure lisibilité et utilisations dans le module de surveillance.

b. Module de surveillance

Le module de surveillance est le deuxième module sur lequel nous avons travaillé. Ce module permet de faire la partie requête SNMP vers le matériel à surveillé. Les informations du matériel à surveillé sont fournies par le module de configuration.



Ce module sera développé en GO. Le choix du langage de programmation GO est dans la philosophie de notre projet en général : simple et rapide. En effet, ce projet est l'occasion d'affirmer nos acquis sur des technologies que nous connaissons (python pour le module précédent), mais c'est aussi le meilleur moment pour découvrir de nouveau langage et de les expérimenter. C'est pour cela que nous nous sommes mis d'accord sur le langage GO. Il est très simple à apprendre dû à ses similitudes syntaxiques avec d'autre langages, ainsi qu'avec sa documentation très fournie. De plus, le langage est très rapide grâce à sa structure et à son système d'héritage et de classe.

Le module comporte plusieurs parties :

- Le script SNMP : ce script va effectuer des requêtes SNMP vers les équipements à surveiller. Pour avoir les informations des équipements, il récupère, via une API, tout ce qui a été renseigné dans le module de configuration. Une fois les requêtes SNMP faites, les données sont envoyées dans la partie suivante.
- La dernière partie est une base de données. Cela sera une base de données InfluxDB et elle stockera toutes les données récupérées sur les équipements. Le choix d'une BDD InfluxDB a été très simple : c'est une base de données écrite en GO (donc en rapport avec l'objectif de faire notre module en GO), optimisé pour du stockage et une lecture rapide, et elle est spécialisée dans le stockage de données métrique (exactement ce que nous récupérerons sur les équipements).

c. Module de Log

Le module de log est le dernier module sur lequel nous allons travailler. Nous ne l'avons pas encore bien défini, c'est pour cela qu'il n'est pas représenté dans le schéma global. Nous avons quand fait une réflexion sur celui-ci : Il devra être léger et rapide, pour respecter la philosophie globale de notre projet. Son objectif est de tracer toutes les informations brutes qui sont récupérées et de les enregistrer dans un fichier consultable par l'administrateur.

III. Planning

Les données indiquées dans le planning sont en « hommes-heures ». Par exemple, pour la première séance, nous avons passés 6 hommes-heures sur le brain storming. Ce qui veut dire que nous avons passé chacun 3h sur cette section.

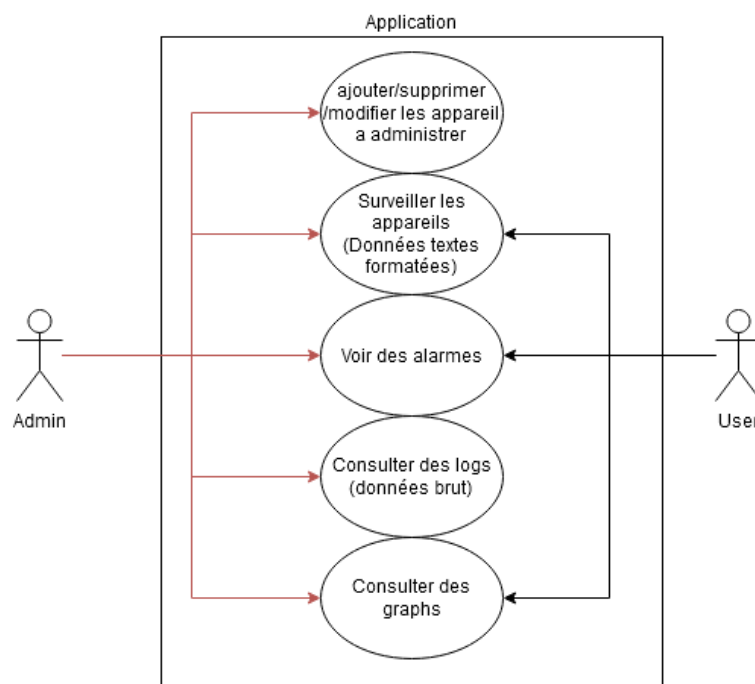
	Séance 1		Séance 2		Séance 3		Séance 4	
	Prévisionnel	Réel	Prévisionnel	Réel	Prévisionnel	Réel	Prévisionnel	Réel
Préparation, brain storming	8	6	1	2				
Module de configuration		2	7	6	2			
Module de surveillance					6	4	4	6
Module de log								
Rapport de projet						4	4	2
	Séance 5		Séance 6		Séance 7		Séance 8	
	Prévisionnel	Réel	Prévisionnel	Réel	Prévisionnel	Réel	Prévisionnel	Réel
Préparation, brain storming								
Module de configuration								
Module de surveillance	8	6	4	4		4		
Module de log			4		8			
Rapport de projet		2		4		4	8	8

	Total d'heure	
	Prévisionnel	Réel
Préparation, brain storming	9	8
Module de configuration	9	8
Module de surveillance	22	24
Module de log	12	0
Rapport de projet	12	24
	64	64

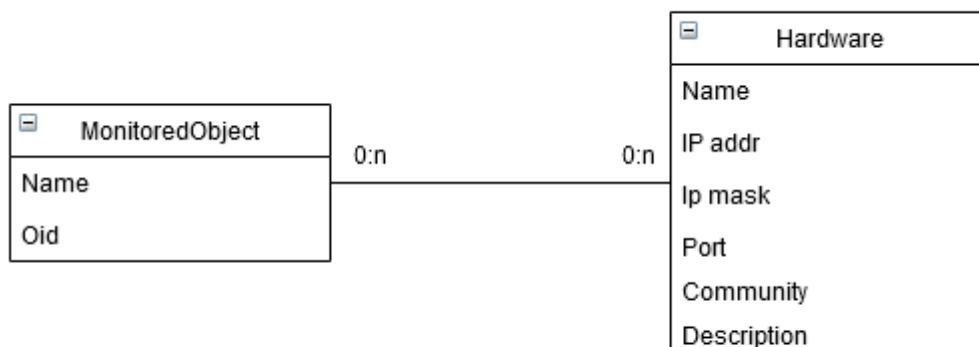
D'après le tableau ci-dessus, nous arrivons à tenir la cadence/objectif fixée au début du projet.

IV. Digrammes

a) Diagramme de cas d'utilisation



b) Diagramme de la base de donnée SQLite



V. Les OID

Le but du projet est de créer un outil de supervision. Pour effectuer nos tests, nous avons décidé d'interroger différents OID. Cela comprend des informations statiques et invariables (ou très peu) comme « sysName » ou « sysContact ». Cela comprend aussi des informations dynamiques qui évoluent comme « snmpInTotalReqVars ». Ci-dessous, une liste des informations, avec leurs OID et la description de celui-ci, que nous avons collectée pour nos tests.

a) Requête GET

Name:	snmplnTotalReqVars
Oid:	1.3.6.1.2.1.11.13
Description:	The total number of MIB objects which have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.
Index:	snmplnTotalReqVars.0

Name:	sysUpTime
Oid:	1.3.6.1.2.1.1.3
Description:	The time (in hundredths of a second) since the network management portion of the system was last re-initialized.
Index :	sysUpTime.0

Name:	sysContact
Oid:	1.3.6.1.2.1.1.4
Description:	The textual identification of the contact person for this managed node, together with information on how to contact this person. If no contact information is known, the value is the zero-length string.
Index :	sysContact.0

Name:	sysName
Oid:	1.3.6.1.2.1.1.5
Description:	An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name. If the name is unknown, the value is the zero-length string.
Index :	sysName.0

Name:	sysLocation
Oid:	1.3.6.1.2.1.1.6
Description:	The physical location of this node (e.g., 'telephone closet, 3rd floor'). If the location is unknown, the value is the zero-length string.
Index :	sysLocation.0

b) Requête snmp Walk

Name:	ifInOctets
Oid:	1.3.6.1.2.1.2.2.1.10
Description:	The total number of octets received on the interface,

VI. Difficultés rencontrées

Nous avons rencontré quelques difficultés lors de la réalisation de notre projet. Ces difficultés ont perturbé notre planning (visible dans la partie planning) ainsi que les tâches prévues.

En l'occurrence, la partie module de surveillance a été la plus longue. Nous avons sous-estimé le temps que demandait cette partie. Aucun problème majeur nous a bloqués, c'était plus des petits problèmes d'implémentation, de connexion, etc...

Une autre chose que nous avons oubliée lors de l'anticipation de notre planning, c'est la dernière séance qui est dédiée aux présentations. Nous avons prévu de travailler totalement sur la présentation et le rapport la dernière séance. Cependant, cette séance est bloquée pour le passage des oraux. De ce fait, cela fait une séance en moins pour travailler, ce qui a amené à répartir sur chaque séance du travail sur le rapport/soutenance, et donc perdre du temps sur la partie développement.

Tous les petits imprévus et ralentissement que nous avons rencontré (qui sont logiques dans le monde du travail) nous a fait revoir très rapidement nos ambitions sur le projet. Pour garantir une application fonctionnelle ainsi qu'un rapport et une présentation de qualité, nous avons décidé de ne pas implémenter le module de log qui, à nos yeux, est la partie-là moins importante du projet.

VII. Amélioration future

Etant donné que le projet a eu une durée limitée, nous avons pensé à des améliorations possibles à apporter à l'application :

- Amélioration et consolidation du code : nous avons fait un code fonctionnel et qui répond aux attentes voulu. Cependant, il est toujours possible de le consolider pour en améliorer les performances ainsi que sa lisibilité.
- Ajout du module de log : à cause à la limite de temps imposé, nous n'avons pas pu implémenter tous les modules. En effet, le module de log a été mis de côté. L'implémentation de ce module dans le futur est plus qu'indispensable.
- Ajout de module supplémentaire : il y a toujours la possibilité d'ajouter de nouveau module pour améliorer l'utilité de l'application au besoin de l'utilisateur.
- Rajouter du contexte aux commandes : actuellement, les commandes sont très simples et statiques. A l'avenir, les commandes pourront être dynamiques. Par exemple, des commandes pourront être automatiquement faites en fonction de l'équipement supervisé (si routeur, alors requête sur la table de route ; si switch, requête sur table de commutation).
- Ajout de sécurité : la sécurité est un élément important et très demandé de nos jours. A l'avenir, le protocole snmpv3 peut être implémenté pour satisfaire ce besoin de sécurité lors du transit des données. De même, un système d'authentification pour l'administrateur et les utilisateurs peut être mis en place.

VIII. Conclusion

Malgré les problèmes de distanciation lié à la crise sanitaire, nous avons quand même réussi à travailler dans de bonnes conditions grâce aux différents outils à notre disposition (visio, GitHub, Drive,...).

Le bon déroulement et suivie du projet est en grande partie dû à une prévision/planification et réflexion mis en place dès son début. Grace à ces bonnes fondations, nous avons pu avancer sans trop de problèmes, car nous avions une vision claire de ce que le produit final devait être.

Nous sommes contents du produit final que nous avons livré. Nous sommes confiants sur le fait que le produit final répond aux problématiques et aux attentes du client.