كلية العلوم و التقنيات فاس
Faculté des Sciences et Techniques de Fès

جامعة سيدي محمد بن عبد الله
Université Sidi Mohamed Ben Abdellah

DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ÉTUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

INTERNET OF THINGS : ÉTAT DE L'ART ET APPLICATIONS

LSSC Laboratoire Signaux Systèmes et Composants

LIEUX DU STAGE : Laboratoire Signaux Systèmes et Composants

Réalisé par :

- Ahmed AMAMOU

Encadré par :

- Pr. Fatiha MRABTI
- Pr. Mohammed OUZARF

Soutenu le 19.06.2019 devant le jury composé de :

- Pr. S. NAJAH	Faculté des Sciences et Techniques de Fès	(Président)
- Pr. L. LAMRINI	Faculté des Sciences et Techniques de Fès	(Examinatrice)
- Pr. F. MRABTI	Faculté des Sciences et Techniques de Fès	(Encadrante)
- Pr. M. OUZARF	Faculté des Sciences et Techniques de Fès	(Encadrant)

Année Universitaire 2018 – 2019

Dédicaces

Je dédie ce modeste travail

A

Mes chers parents

En témoignage de ma reconnaissance envers le soutien, les sacrifies et tous les efforts qu'ils ont fait pour mon éducation ainsi que ma formation

A

Mes sœurs, mes amis et collègues pour

Leur encouragement, leurs aides et leurs patiences au cours de mes années d'études ;

A

Tous les enseignants du Département Informatique

Leur générosité et leur soutien m'oblige de leurs témoigner mon profond respect et ma loyale considération.

Remerciement

Dieu merci pour la santé, la volonté, le courage et la détermination qui m'ont accompagné tout au long de la préparation et l'élaboration de ce travail.

Je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce travail ainsi qu'à la réussite de cette formidable année universitaire.

Ces remerciements vont tout d'abord au corps professoral et administratif de la Faculté des Sciences et Techniques de Fès, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

Je tiens à remercier sincèrement **Pr. Fatiha MRABTI** et **Pr. Mohammed OUZARF**, qui en tant que encadrants, se sont toujours montrés à l'écoute et très disponible tout au long de la réalisation de ce travail, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu nous consacrer et sans qui ce travail n'aurait jamais vu le jour.

Mes vifs remerciements vont également aux membres du jury **Pr. Said NAJAH** et **Pr. Loubna LAMRINI** pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner et de l'enrichir par leurs propositions.

Je n'oublie pas mes parents pour leur contribution, leur soutien et leur patience, et **Mr. El Mostafa SEBAAOUI** pour m'avoir épaulé moralement tous les jours dans la construction de ce travail et ses conseils durant toutes mes années d'études.

Enfin, je m'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours encouragée au cours de la réalisation de ce travail.

Merci à tous et à toutes.

Résumé

L'Internet of things (IoT) rend les objets qui nous entourent intelligents en les connectant via un réseau. L'IoT est un domaine d'actualité qui forme un écosystème d'objets, de communications, d'applications et d'analyses des données.

Dans ce travail nous avons essayé d'une part de définir l'IoT, présenter son architecture, ses domaines d'applications, ses défis et les différents travaux de recherche. La sécurité représente un problème majeur de l'IoT, nous avons donné ses problématiques, les techniques utilisées et aussi ses défis.

D'autre part nous avons configuré un serveur Dell PowerEdge R440 pour utilisation futur de développement d'application IoT, puis nous avons comparé ses performances avec d'autres ordinateurs.

Nous avons aussi développé une application de mesure de température à l'aide d'ordinateur monocarte PhidgetSBC3 et un capteur de température. Cette application nous donne la température pour chaque seconde dans une durée déterminée par l'utilisateur.

Mots clés : internet of things, Dell PowerEdge

Abstract

The Internet of things (IoT) makes objects around us smart by connecting them via a network. IoT is a topical field that forms an ecosystem of objects, communications, applications and data analysis.

In this work we tried on the one hand to define IoT, to present its architecture, its fields of applications, its challenges and the different works of researchers. Security is a major problem of IoT, we gave its problems, the techniques used and also its challenges.

On the other hand, we configured a Dell PowerEdge R440 server for future use in IoT application development, and then compared its performance with other computers.

We have also developed a temperature measurement application using a PhidgetSBC3 monocarte computer and a temperature sensor. This application gives us the temperature for each second within a duration determined by the user.

Keywords: internet of things, Dell PowerEdge

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

2D	Deux Dimensions
ADM	Automating desin methodology
ADS	dans les systèmes distribués artificiels
API	application programming interface
BIOS	Basic Input Output System
CIA	Confidentiality integrity availability
CoAP	Constrained Application Protocol
CPS	Cyber Physical Systems
CPU	Central Processing Unit
CRC	Cyclic redundancy check
CSV	Comma-separated values
DoS	Disk operating system
EDBO	Emergent Distributed Bio-Organization
EIS	systèmes Internet embarqués
GPRS	General Packet Radio Service
GW	Gateway
H3IoT	Home Health Hub Internet of Things
HDD	Hard Disk Drive
HPC	High performance computing
HTML	HyperText Markup Language
IAL	Internet Application Layer
ID	Identifiant
IERC	European Research Cluster on the Internet of Things.
IoT	Internet of things
IPL	Information Processing Layer
Json	JavaScript Object Notation
LCL	Local Communication Layer
m-learning	Mobile Learning
NumPy	Numeric Python
PSL	Physiological Sensing Layer
RAID	Redundant Arrays of Inexpensive Disks

RAM	Random Access Memory
RDD	Resilient Distributed Datasets
RFID	Radio Frequency identification
SCA	Attaque sur le canal latéral
SDD	Solid-state drive
SENSAPP	Application cloud prototypique à code source ouvert
SOA	Service Oriented Architecture
UAL	User Application Layer
URI	Uniform Resource Identifier
WAN	Wide Area Network
WiFi	Wireless Fidelity
WSN	Wireless Sensor Networks

Introduction générale

Le terme **I**nternet **o**f **T**hings (IoT) est apparu en 1999 par Kevin Ashton pour décrire des objets équipés de puces RFID. Depuis l'IoT ne cessa de s'évoluer et elle s'est généralisée vers un concept connectant un grand nombre d'objets pour collecter des informations et offrir des services.

D'ici 2020, l'Institut Gartner prévoit plus de 50 milliards d'objets connectés sur le marché. C'est une révolution technologique qui change notre style de vie. L'IoT est capable d'interagir sans intervention humaine. Certaines applications préliminaires de l'IoT ont déjà été développées dans différents domaines : la santé, les transports et l'éducation... Pour atteindre son potentiel, des mesures doivent être prises, plus spécialement l'aspect de l'architecture et de la sécurité qui sont des préoccupations majeures de l'IoT.

Dans le cadre de notre travail, nous avons établi une étude bibliographique sur l'IoT, ses problématiques, les travaux de recherche et sa sécurité. Nous avons aussi configurer un serveur du laboratoire LSSC, comparé ses performances par rapport à d'autres ordinateurs, et développé une application IoT de mesure de température à l'aide d'un ordinateur monocarte et un capteur de température.

Ce rapport est organisé de la manière suivante :

Dans le premier chapitre nous avons commencé par la définition de l'IoT, son architecture, ses domaines d'application, ses défis et les différents travaux effectués par les scientifiques.

Dans le deuxième chapitre, nous nous sommes focalisés sur la sécurité, en emmenant une étude sur l'état de l'art, en dégageant les problématiques, les techniques utilisées et les défis.

Le dernier chapitre, est consacré à la configuration du serveur Dell PowerEdge R440, le résultat de comparaison de ce serveur par rapport à d'autres ordinateurs, ainsi qu'une comparaison entre trois librairies python de traitement big data. Une application IoT de mesure de température.

Chapitre I: Etat de l'art

IoI

IoT (Internet of things) est un réseau mondial d'objets connecté entre eux grâce à internet, ces objets sont adressables de manière unique. Tous les objets sont capables d'émettre de l'information et de recevoir des commandes. IoT ouvre la voie vers une multitude de scénarios basés sur l'interconnexion entre le monde physique et le monde virtuel.

Dans ce chapitre, nous allons citer quelques définitions de l'IoT et donner son architecture, ses domaines d'applications et ses Défis. Nous allons présenter, par la suite, des travaux de recherches.

I.1 Problématiques

L'IoT est une évolution d'un réseau d'objets interconnectés. D'une vision technologique à la réalité et ses contraintes, l'IoT fait face à plusieurs problématiques. Dans cette section nous allons présenter l'IoT, son architecture, ses domaines d'applications et ses défis. Ainsi que les travaux de recherches.

I.1.1 Définitions

Il existe plusieurs définitions de l'IoT, des définitions insistent sur les aspects techniques, et d'autres portent sur les usages et les fonctionnalités, nous allons citer quelques-unes :

Le groupe RFID (radio frequency identification) définit l'Internet des objets comme étant le réseau mondial d'objets interconnectés, pouvant être adressés de manière unique en fonction de protocoles de communication standard.[1]

L'internet des objets est aussi l'interconnexion de dispositifs de détection et d'actionnement offrant la possibilité de partager des informations sur plusieurs plates-formes via un cadre unifié, développant ainsi un tableau de fonctionnement commun permettant des applications innovantes Ceci est réalisé grâce à une détection omniprésente, à l'analyse de données et à la représentation d'informations, le cloud computing servant de cadre unificateur.[1]

Selon le groupe de projets de recherche européens sur l'Internet des Objets (IERC-European Research Cluster on the Internet of Things.) : Les objets sont des participants actifs à des processus commerciaux, informatiques et sociaux. Ils sont capables d'interagir et de communiquer entre eux et avec l'environnement en échangeant des données et des informations relatives à ce dernier, tout en réagissant de manière autonome aux événements du monde réel et physique et en influençant celui-ci. Ceci en exécutant des processus qui déclenchent des actions et créent des services avec ou sans intervention humaine directe.[2]

I.1.2 Architecture

IoT désigne un ensemble objets communicants et connectés, des réseaux de télécommunication, des transmissions de données, de l'hébergement et du stockage de données et surtout de l'intelligence applicative permettant de donner de la valeur à la donnée collectée.

L'architecture d'un système IoT est centralisée, composée de plusieurs niveaux qui communiquent entre eux pour relier le monde tangible des objets au monde virtuel des réseaux. Tous les projets n'adoptent pas une architecture formellement identique, néanmoins il est possible de schématiser le parcours de la donnée.

Les données doivent être stockées et utilisées intelligemment. Il est important de développer des algorithmes d'intelligence artificielle qui pourraient être centralisés ou distribués en fonction des besoins. Donc l'architecture représente un grand défi pour l'IoT[1]

On distingue deux types d'objet :

- **Les objets passifs :** ils utilisent généralement un tag (puce RFID, code barre 2D). Ils ont une capacité de stockage faible (de l'ordre du kilooctet) et permettent de jouer le rôle d'identification. Ils peuvent parfois, dans le cas d'une puce RFID, d'embarquer un capteur (température, humidité) et être réinscriptibles.
- **Les objets actifs :** ils peuvent être équipés de plusieurs capteurs, avec une grande capacité de stockage, capables d'accomplir des calculs et être en mesure de communiquer sur un réseau.

Pour IoT dans une perspective de haut niveau, nous décrivons l'architecture de l'IoT en trois couches :

- **Matériel:** composé de capteurs, d'actionneurs et de matériel de communication intégré.
- **Middleware :** Stockage à la demande et outils informatiques pour l'analyse des données.
- **Présentation:** nouveaux outils de visualisation et d'interprétation faciles à comprendre, facilement accessibles sur différentes plates-formes et conçus pour différentes applications.[1]

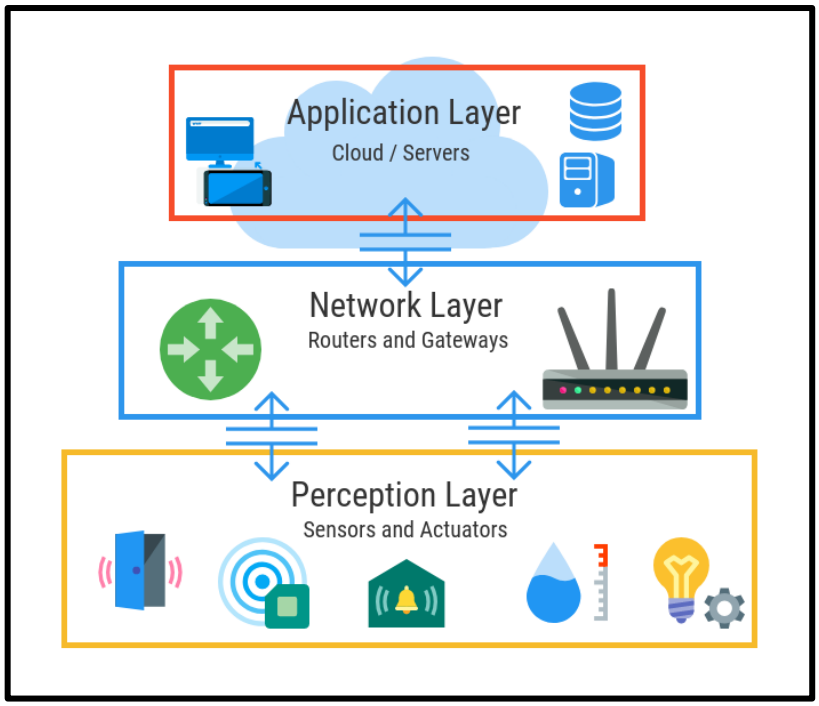

Figure 1 : Exemple d'une architecture IoT

Parmi les éléments qui composent ces trois couches, on trouve :

I.1.2.1 Devices

I.1.2.1.1 Radio Frequency Identification (RFID)

Permet la conception de micro puces pour la communication de données sans fil. Ils aident à l'identification automatique de tout ce à quoi ils sont attachés en agissant comme un code à barres électronique.[3]

Les étiquettes RFID passives ne sont pas alimentées par batterie et utilisent la puissance du signal d'interrogation du lecteur pour communiquer l'ID au lecteur RFID.[3]

Les lecteurs RFID actifs disposent de leur propre batterie et peuvent instancier la communication. [3]

Figure 2 : Etiquette RFID

I.1.2.1.2 Wireless Sensor Networks (WSN)

Les composants qui constituent le réseau de surveillance WSN sont les suivants:

- **Matériel WSN**: un nœud (matériel principal WSN) contient généralement des interfaces de capteur, des unités de traitement, des émetteurs-récepteurs et une alimentation.[4]
- **Pile de communication WSN**: les nœuds doivent être déployés de manière ad hoc pour la plupart des applications. La pile de communications au niveau du nœud collecteur devrait pouvoir interagir avec le monde extérieur via Internet pour servir de passerelle vers le sous-réseau WSN et Internet.[5]
- **WSN Middleware**: un mécanisme permettant de combiner une infrastructure informatique avec une architecture SOA (Service Oriented Architecture) et des réseaux de capteurs afin de fournir un accès à des ressources de capteurs hétérogènes de manière indépendante du déploiement. Ceci est basé sur l'idée d'isoler des ressources pouvant être utilisées par plusieurs applications.[5]
- **Agrégation de données sécurisée** - Une méthode d'agrégation de données efficace et sécurisée est nécessaire pour prolonger la durée de vie du réseau et garantir la fiabilité des données collectées à partir de capteurs.[6]

I.1.2.2 Communication

I.1.2.2.1 Adressage

L'identification unique d'un objet est un grand défi pour IoT, en plus d'identifier plusieurs objets, elle nous permet de les contrôlés via internet. Chaque élément déjà connecté et ceux qui vont l'être doivent être identifiés par leurs identifications, emplacements et fonctionnalités uniques.

L'ajout des réseaux et d'appareils ne doit pas dégrader les performances du réseau, le fonctionnement des appareils, la fiabilité des données sur le réseau ou l'utilisation efficace des appareils à partir de l'interface utilisateur.

I.1.2.2.2 Visualisation

La visualisation est essentielle pour une application IoT, car elle permet l'interaction de l'utilisateur avec l'environnement. Pour faciliter le développement d'une application IoT, on utilise une architecture à trois couches :

- **Une couche intégrée**, mise en œuvre sous la forme d'un objet physique agrémenté de capteurs, d'actionneurs et d'une connectivité sans fil à courte portée pour fournir des capacités de détection et d'interface utilisateur;[7]

- **Une couche passerelle**, mise en œuvre sous la forme d'un périphérique tel qu'un smartphone ou un routeur WiFi, afin de fournir une connectivité à la couche intégrée, permettant ainsi un accès omniprésent à l'information;[7]
- **Une couche serveur**, mise en œuvre en tant que service cloud, qui permet le stockage de données et l'intégration avec des services tiers.[7]

I.1.2.3 IoT Cloud

La vision de IoT peut être vue sous deux angles: centré sur « Internet » et centré sur « Thing ».

L'architecture centrée sur Internet mettra l'accent sur les services Internet, tandis que les objets fourniront des données. [1]

Dans l'architecture centrée sur les objets, les objets intelligents occupent une place centrale.

Pour la première approche centrée sur internet, Un cadre conceptuel intégrant les dispositifs de détection omniprésents et les applications est présenté dans Figure qui suit.

Les fournisseurs de services de détection peuvent rejoindre le réseau et proposer leurs données via un nuage de stockage; les développeurs d'outils analytiques peuvent fournir leurs outils logiciels.

Les experts en intelligence artificielle peuvent fournir leurs outils d'exploration de données et d'apprentissage automatique utiles pour la conversion de l'information en connaissances. Enfin, les concepteurs en infographie peuvent proposer divers outils de visualisation.

Le cloud computing peut offrir ces services en tant qu'infrastructures, plates-formes ou logiciels, où tout le potentiel de la créativité humaine peut être exploité en les utilisant comme services.

I.1.3 Domaines d'application

IoT permet le développement de plusieurs applications intelligentes qui touchent essentiellement : la domotique, les villes, le transport, la santé et l'industrie. Dans ce qui suit, nous citons brièvement des exemples du domaine d'applications de IoT.

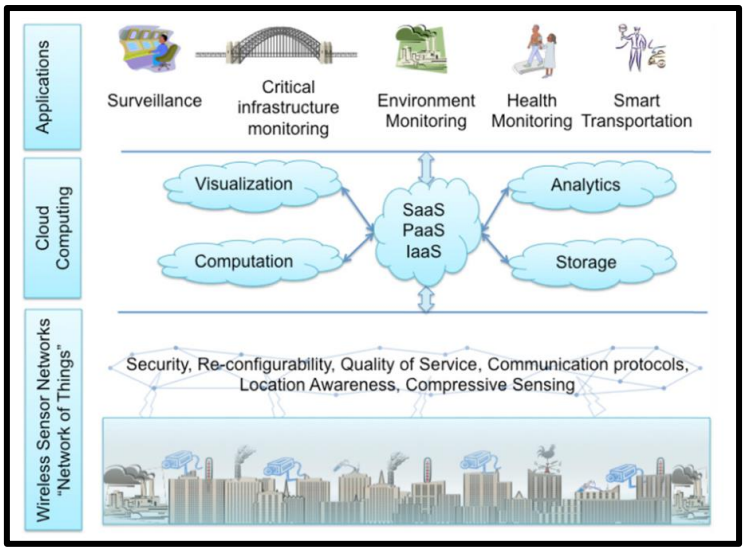

Figure 3 : Exemple d'une architecture IoT

I.1.3.1 La santé

Les hôpitaux mondiaux utilisent déjà l'Internet des Objets pour augmenter leur productivité et améliorer les soins apportés aux patients.

Les objets connectés sont utilisés au quotidien pour :

- La surveillance au sein des établissements médicaux et la maintenance
- Les opérations chirurgicales et le contrôle à distance
- Les services de géolocalisation

I.1.3.2 La domotique ou maison connectée

La maison intelligente est en train de se normaliser.

Outre les objets de divertissement comme les télés intelligentes ou les enceintes connectées, la domotique a pensé également à la sécurité et l'économie d'énergie au sein de l'habitat :

- Centrale domotique : contrôle et programmation de différentes interventions à l'intérieur du foyer
- Capteurs d'informations (système d'alarme, variations de température, etc.)
- Actionneurs, qui permettent la programmation et le contrôle des différents appareils électroniques du foyer, même à distance

I.1.3.3 L'industrie connectée

L'industrie n'est pas en reste sur l'usage de l'Internet des Objets et des bénéfices que celui-ci lui apporte. L'usage des objets connectés est très spécifique et répond à des besoins :

- D'optimisation (chaîne logistique)
- De transformation des processus d'entreprise
- D'amélioration de l'efficacité et de la productivité
- De traçabilité et de sécurité

I.1.3.4 L'agriculture

L'IoT est utilisé pour récolter en temps réel des informations essentielles à la gestion de l'exploitation :

Humidité de la terre, état des plantations, climat, etc.

Les données récoltées sont transférées aux tracteurs connectés (parfois autonomes). Cela permet de doser finement le niveau d'engrais et d'arrosage sur telle ou telle parcelle et de réduire les coûts, tant financiers qu'énergétiques.

I.1.3.5 L'élevage

Les objets connectés ne sont pas seulement utiles aux agriculteurs, mais également aux éleveurs qui peuvent surveiller plus finement l'état de santé de leurs bêtes. Aussi des traceurs GPS pour le recueillement des habitudes alimentaires des bovins,

I.1.3.6 Smart retail : des supermarchés branchés

La technologie d'identification par radiofréquence (RFID) permet de renforcer l'expérience client en offrant un parcours client ultra personnalisé. Outre les applications mobiles, des concepts de caddies connectés ont déjà été pensés pour faciliter les courses en supermarché :

- Liste de courses intégrée
- Parcours guidé pour optimiser le temps de course
- Calcul automatique du montant du panier, …

Les commerçants investissent également dans les applications mobiles pour fidéliser et attirer les clients vers les boutiques physiques, par le biais, par exemple, de notifications sur les promotions / soldes en cours, lors du passage d'un client près d'une boutique.

I.1.3.7 Des villes intelligentes et connectées

Appelées « smart cities », les villes ont entamé leur transition digitale pour répondre aux enjeux de la société moderne. Urbanisme, économie et développement durable forment des enjeux importants pour les villes de demain, qui doivent répondre aux besoins d'une population toujours plus dense avec des ressources de plus en plus limitées.

Là encore, IoT est utile :

- Foyers domotiques
- Supports numériques
- Capteurs et compteurs intelligents
- Bornes de rechargement pour véhicules électriques
- Éclairage citadin intelligent, etc.

I.1.3.8 La sécurité routière

Particulièrement populaire depuis ces dernières années, la voiture connectée participe grandement au renforcement de la sécurité routière. La révolution numérique a offert à l'industrie automobile des perspectives jusqu'alors jamais vues.

- Boitier d'appel d'urgence autonome
- Tableau de bord synchronisé avec le smartphone
- Développement d'applications sur les plateformes dédiées…

I.1.4 Défis

L'Internet des objets est un réseau d'objets interconnectés. Pour permettre à ce réseau d'atteindre son potentiel, plusieurs aspects doivent être étudiés et nécessitent de résoudre un certain nombre de problématiques :

I.1.4.1 Interopérabilité technologique

L'interopérabilité est beaucoup plus difficile pour l'IOT, car il ne s'agit pas uniquement de connecter des personnes avec des personnes, mais d'une interaction transparente entre les appareils et les personnes possédant des appareils. Ces appareils peuvent différer quant à leurs capacités technologiques.[8]

En plus, l'accès à des éléments de réseau hétérogène (Internet, les réseaux de télécommunication, …) à grande échelle et l'échange massif de données entre eux sont les nouvelles caractéristiques importantes associées à la large application de l'IoT.

Le défi posé alors est l'interconnexion hautement efficace entre les éléments de ce réseau hétérogène à grande échelle qui est défini comme un problème scientifique clé de l'IOT.[9]

I.1.4.2 La Sécurité des données

Une grande partie des données acquises et communiquées contiennent des informations personnelles. Le défis de l'IoT c'est d'assurer l'intégrité des données et leurs cryptages pour les protéger d'éventuelles attaques, la propriété des données et les formalités juridiques.[8]

L'utilisation massive des réseaux sans fil facilite les écoutes clandestines, même en l'absence d'intentions malveillantes de la part des propriétaires ce qui permet de dresser des profils très détaillés de propriétaires de ces objets et de les suivre à grande échelle..[10]

I.1.4.3 Consommation d'énergie

Un objet connecté doit rester autonome pour communiquer, et cela coute de l'énergie. Minimiser la consommation d'énergie représente un défi majeur de l'IoT.[8]

I.1.4.4 Adressage et nommage

Le très grand nombre d'objets nécessite un espace d'adressage important et augmente significativement la quantité d'informations que les serveurs de noms doivent stocker pour assurer leur rôle d'association entre les noms d'objet et leurs adresses.[10]

I.1.4.5 Flux de données

Les informations produites par les capteurs sont naturellement liées au temps, sous la forme de flux de mesures ou d'évènements. Cette particularité s'oppose radicalement aux approches classiques basées sur des ensembles de données finis, et nécessite une réflexion différente en ce qui concerne la représentation des données (data model) et leur traitement (computation model). Sans oublié les informations erronées envoyées par les capteurs, il faut aussi utiliser des techniques de détection et correction de ses erreurs.[10]

I.1.4.6 Formalisation

Pour obtenir des données complètes et fiables, les nœuds intelligents et les équipements RFID doivent être disposés massivement et raisonnablement. En ce qui concerne la couche de détection, lors d'une panne partielle, les autres composants doivent être remplacés.[11]

I.1.4.7 Standardisation

La technologie RFID, le protocole de communication et la couche application doivent tous être normalisés. À l'heure actuelle, il n'existe pas de technologie standard d'unification RFID dans le monde. Pour promouvoir l'application IOT, nous devons accélérer la normalisation internationale RFID.[11]

Dans la section suivante, nous donnons quelques travaux de recherche dans l'IoT.

I.2 Production scientifique

L'IoT ne cesse d'évoluer, elle touche désormais tout notre quotidien, mais cela n'est pas facile car elle fait face aux plusieurs défis principalement l'architecture et la sécurité.

Plusieurs chercheurs ont effectué des travaux sur l'IoT, cette section décrit les travaux réalisés jusqu'à présent par les scientifiques du monde entier dans diverses architectures spécialisées basées sur les grands domaines.

I.2.1 Architecture

Plusieurs travaux ont été effectués par les chercheurs sur les différentes architectures de IoT, tels que:

I.2.1.1 L'architecture orientée services

L'architecture orientée service (SOA), est une approche utilisée pour créer une architecture basée sur l'utilisation de services système.

L'approche SoA intégrée est actuellement invoquée dans le domaine de l'IoT, utilisant le concept de middleware, c'est-à-dire une couche logicielle superposée entre la couche application et la couche technologie qui cache les détails inutiles et pertinents du produit développé, réduisant ainsi le temps de développement du produit, facilitant le flux de travail de conception. [12]

I.2.1.2 RFID

Les chercheurs ont mis au point un réseau de capteurs RFID-SN comprenant une étiquette RFID, un lecteur et un système informatique permettant de comprendre le comportement du système [13].

Fosstrak One (Free and Open Source Sofwtare for Track and Trace) a développé une nouvelle application liée à la RFID basée sur la gestion SoA (Foss Track). Les scientifiques ont proposé un système basé sur un lecteur RFID configuré par réseau EPA en fournissant plusieurs services liés aux données sur sa couche applicative, par exemple agrégation, filtrage, service de recherche et d'annuaire, gestion des identificateurs de balises et confidentialité, en utilisant le paradigme SoA.[14]

I.2.1.3 Middleware

Une architecture middleware à 3 couches basée sur la RFID repose sur trois fonctionnalités associatives telles que: l'association de balises, l'association de lieu et l'association d'antenne avec l'utilisateur.[15].

Une architecture holistique IoT est proposée et comprend des périphériques hétérogènes, des systèmes Internet embarqués (EIS), des protocoles de communication standard et un paradigme SoA qui utilise le protocole CoAP et des services standard en permettant l'échange de données de capteurs avec un cloud IoT et un cloud privé, tout en diffusant une interface homme-machine basée sur le Web pour la configuration, la surveillance et visualisation de données de capteurs structurés.[16]

La plate-forme INOX préconise une approche similaire composée de trois couches, à savoir: (a) couche de service - prend en charge et contient les services à l'aide d'API, (b) couche de plate-forme - contient la gestion et l'orchestration nécessaires au déploiement de services et les technologies de virtualisation enrichissant la couche matérielle ; et (c) Couche matérielle - contient des capteurs et des objets intelligents. [17]

I.2.1.4 Réseau de capteurs distribués

Le modèle Emergent Distributed Bio-Organization (EDBO),[18], est conçu pour exploiter les phénomènes émergents dans les systèmes distribués artificiels (ADS). Les nœuds EDBO sont représentés par des agents « BioBots » qui utilisent des relations bidirectionnelles pour former un réseau de superposition. Chaque BioBot est capable de gérer un nombre limité de relations avec d'autres BioBots dans un environnement autonome.

BioBot sert d'enveloppeur pour l'abstraction, les données, les fonctionnalités et les services basés sur les demandes de l'utilisateur. Il facilite la propagation des requêtes à travers le réseau de manière autonome où son comportement est basé sur plusieurs mécanismes heuristiques bio-inspirés qui aident à participer à la prise de décision.

L'architecture tire parti de la combinaison de plusieurs BioBot habilités par des nœuds de système physique cybernétique (CPS) positionnés dans des emplacements distribués. Les utilisateurs peuvent invoquer leurs requêtes sur l'EDBO qui est ensuite traitée par des décisions collectives prises par le BioBot avec l'intervention de nœuds CPS.

I.2.2 Domaines d'utilisations

I.2.2.1 Agriculture

L'IoT agricole est envisagé en développant une plate-forme prototype qui contrôle l'intégration de l'information de réseau pour étudier la situation réelle de la production

agricole tout en opérant à distance. Cette étude utilise WSN comme base de la mise en œuvre.[19]

Un travail récent a proposé une architecture agricole à six niveaux qui incorpore WSN comme un élément subsidiaire pour améliorer l'analyse multi-culture, l'expérience utilisateur et l'analyse prédictive.[20]

I.2.2.2 Santé

Récemment, le développement et la diffusion de systèmes de santé intelligents sont devenus possibles grâce à la convergence de diverses architectures de l'IoT.

iHome Health-IoT est une plate-forme pour des services de soins de santé basés sur l'IoT; illustrant une boîte de médecine intelligente (iMedBox) à 3 niveaux à plateforme ouverte pour poursuivre diverses installations médicales intégrées avec des capteurs et appareils, et communiquer au moyen du WAN, du GPRS et/ou de la 3G.

Les services, comme l'emballage pharmaceutique intelligent (iMedPack), sont activés par RFID et la capacité d'actionnement est activée par des matériaux fonctionnels, flexible, et portable dispositif de capteur biomédical (Bio-Patch) [21].

Biopatch prend la décision d'appeler un médecin éloigné, un centre d'urgence, un hôpital, une clinique d'essai et des détaillants en médecine de la chaîne d'approvisionnement.

Home Health Hub Internet of Things (H3IoT) [22], est une plate-forme conçu pour diffuser les soins de santé des personnes âgées à domicile. Il s'agit d'une approche à cinq niveaux (c.-à-d. couche de détection physiologique (PSL), couche de communication locale (LCL), couche de traitement de l'information (IPL), couche d'application Internet (IAL) et couche d'application utilisateur (UAL).) pour évaluer et surveiller les changements physiologiques chez les personnes âgées et pour prendre des mesures subséquentes pour un examen de santé plus poussé par le médecin et les donneurs d'acres.

L'architecture de l'hôpital intelligent fondée sur l'IoT, est mise en œuvre pour améliorer l'efficacité du système d'information actuel des hôpitaux, comme le point d'information fixe, le mode de fonctionnement du réseau exigible et les paramètres connexes.[23]

Le système de conception automatisée (ADM) a été conçu pour la réadaptation intelligente des personnes âgées. Ce type de plateforme basée sur l'ontologie crée une stratégie de réadaptation et reconfigure les ressources médicales en fonction des besoins spécifiques des patients rapidement et automatiquement.[24]

I.2.2.3 Smart City

La surveillance de l'état des routes et la génération d'alertes ont été effectuées en utilisant le Smartphone embarqué comme capteurs connectés à une plate-forme IoT. [25]

Le réseau de véhicules utilisant des middlewares basés sur l'IoT a été introduit pour gérer efficacement les véhicules routiers.[26]

Une expérience de smart city décrit l'architecture d'expérimentation et le déploiement de l'IoT dans la ville Santander. La même a été présenté comme une architecture à trois niveaux : un dispositif terminal, une passerelle (GW) et un serveur pour faciliter l'infrastructure de Smart Santander. [27]

Un système de gestion des urgences basé sur l'IoT a été proposé qui traite les événements catastrophiques d'une manière spécialisée.[28]

Le SENSAPP a été conçu comme une application cloud prototypique à code source ouvert pour stocker et exploiter les données recueillies par l'Internet des objets. L'architecte de capteurs et le logiciel de mineur de données traitent les données de l'IoT recueillies à partir du capteur attaché à un vélo. La base de données et le système font face aux tâches liées à la notification. L'utilisateur peut facilement accéder à l'information et l'utiliser à distance à l'aide d'un logiciel tiers.[29]

Une architecture générique de l'Internet des objets pour le sport intelligent « Internet des objets sport » a été proposée pour faciliter les interactions intégrées entre les sportifs, le sport objets, propriétaire de l'équipe, équipes médicales et adeptes.[30]

I.2.2.4 Apprentissage

Un modèle fonctionnel est proposé pour répondre aux besoins de l'apprentissage mobile futuriste (m-learning) par l'Internet des objets. Ce modèle peut être exploité par quatre facteurs, notamment :

(a) Créer un environnement d'apprentissage optimal pour le m-learning,

(b) fournir des ressources de masse pour le m-learning,

(c) rendre service individuel de m-learning,

(d) enrichir la méthode d'évaluation.

Le mode m-learning, basé sur l'architecture de l'Internet des Objets (IOT-ML) est conçus pour effectuer plusieurs tâches comme : l'analyse préliminaire, la création de situation d'apprentissage, l'acquisition de ressources d'apprentissage, et l'évaluation de l'infrastructure d'apprentissage par une rétroaction rigoureuse et un environnement d'apprentissage axé sur la poussée et la traction.[31]

I.2.2.5 Big Data

Les architectures IoT génèrent différents types de données en grand volume à très grande vitesse. Un cadre de stockage de données a été proposé [32], intégrant des données structurées et non structurées.

La nouvelle architecture combine « Hadoop » et de multiples autres bases de données, pour créer un dépôt distribué de données pour stocker et gérer de façon efficace divers types de données recueillies par les capteurs et les lecteurs d'IRF.

L'architecture en 6 couches place des dispositifs hétérogènes qui résident en bas alors que les systèmes de base de données, tels que : Hadoop, NoSQL, et base de données Relational couvrent la couche supérieure suivante. Les dépôts de données et de données sont placés au-dessus de celui-ci, tirant parti de la gestion multi-locataires et de versions, de la cartographie des objets ainsi que de la connectivité des bases de données.

La couche de gestion des services fournit des activités de génération de services en plus des RESTful API et des URI. La couche supérieure la plus importante est la couche d'application qui répond à l'expérience directe de l'utilisateur avec le contenu collecté par les appareils et représenté de manière orientée vers la connaissance aux utilisateurs.

Pour gérer les données générées par un grand nombre de capteurs dans un système basé sur l'IoT, on aborde les paramètres contextuels ainsi que les données particulières pour lesquelles l'analyse est nécessaire. Une architecture proposée regroupe l'approche de l'altération, de la transformation et de l'intégration des données.[33]

I.2.2.6 Sécurité

Pour que l'IoT puisse déployer tout son potentiel, il est essentiel de gagner et de conserver la confiance des utilisateurs en matière de confidentialité et de sécurité. En effet, les transferts de données liés à l'IoT traceront le portrait de chacun d'entre nous. L'enjeu est de sécuriser ces informations.

De nombreux moyens sont à la disposition d'un hackeur pour accéder aux fonctionnalités ou aux données d'un objet connecté. Les trois principales cibles de « piratage » sont les suivantes : l'objet connecté, le réseau et les serveurs. Nous allons présenter dans le chapitre suivant les différentes problématiques de la sécurité, les techniques déjà utilisées, et les défis.

Jusqu'à présent, elle n'existe aucune définition standard pour l'IoT, l'IoT est définie soit par son architecture soit par ses fonctionnalité, même pour l'architecture elle n'existe aucun standard, mais l'architecture à trois couches peut être considérer comme référence. L'IoT touche tous les domaines mais elle doit faire face à plusieurs défis.

Chapitre II: Sécurité

La sécurité représente un grand défi pour l'IoT.

La triade Sécurité ou CIA, est un modèle distingué pour le développement de mécanismes de sécurité, qui met en œuvre la sécurité en utilisant les trois zones qui sont la confidentialité, l'intégrité et la disponibilité des données.

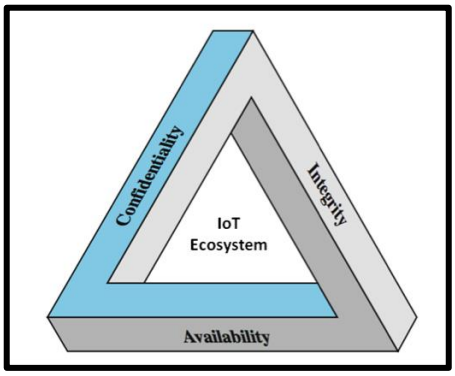

Figure 4 : La triade Sécurité CIA

Dans ce chapitre nous allons voir les problématiques de la sécurité dans l'IoT, les techniques utilisées ainsi que ses défis.

II.1.1 Problématiques

Une faille dans l'une des zones du triade Sécurité pourrait causer de graves problèmes au système. Les trois zones sont décrites ci-dessous :

II.1.1.1 Confidentialité

C'est la capacité de donner confiance à l'utilisateur sur la confidentialité des informations sensibles. Il existe de nombreux mécanismes de sécurité pour assurer la confidentialité des données, par exemple :

Le cryptage des données, dans lequel les données sont converties sous forme de texte crypté, ce qui rend difficile l'accès aux utilisateurs n'ayant pas les autorisations appropriées.

La vérification en deux étapes, qui fournit une authentification par deux composants dépendants et autorise l'accès uniquement si les deux composants réussissent le test d'authentification on note la vérification biométrique est plus courante dans laquelle chaque personne est identifiable de manière unique. [34]

II.1.1.2 Intégrité

L'intégrité des données fait référence à la protection d'informations utiles contre les cybercriminels ou aux interférences externes lors de la transmission et de la réception à l'aide de méthodes de suivi communes, de sorte que les données ne puissent pas être falsifiées sans que le système ne détecte la menace.[35]

Parmi les méthodes utilisées pour assurer l'intégrité des données, on trouve CRC, une synchronisation continue des données à des fins de sauvegarde et une fonction telle que le contrôle des versions, qui enregistre les modifications de fichier dans un système pour restaurer le fichier en cas de suppression fortuite de données.

II.1.1.3 Disponibilité

La disponibilité des données garantit l'accès immédiat des parties autorisées à leurs ressources d'informations non seulement dans les conditions normales, mais également dans des conditions désastreuses.

La disponibilité des données permet également d'éviter les goulots d'étranglement empêchant la circulation de l'information. Les méthodes de sauvegarde par redondance et par basculement permettent de dupliquer les composants du système en cas de défaillance du système ou de conflits divers afin de garantir la fiabilité et la disponibilité des données.[36]

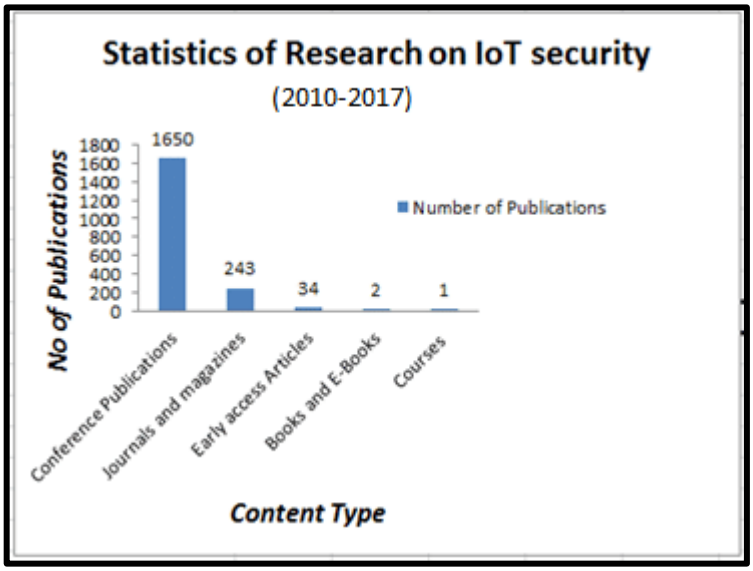

Figure 5 : Statistiques des travaux de recherche dans la sécurité IoT de 2010 à 2017

II.1.2 Les techniques de sécurité existantes

II.1.2.1 Perception Layer :

- **Authentification** : L'authentification est effectuée à l'aide d'algorithmes de hachage cryptographique, qui fournissent des signatures numériques aux terminaux capables de résister à toutes les attaques connues telles qu'une attaque par canal latéral, une attaque par force brute et une attaque par collision, etc.[36]
- **La confidentialité des données** : La confidentialité des données est garantie par des algorithmes de chiffrement symétriques et asymétriques tels que RSA, DSA, BLOWFISH et DES, etc., qui empêchent un accès non autorisé aux données du capteur

lors de la collecte ou du transfert à la couche suivante. En raison de leur faible consommation d'énergie, ils peuvent être facilement implémentés dans les capteurs.[37]

- **Confidentialité des informations sensibles** : En ce qui concerne le masquage des informations sensibles, l'anonymat de la localisation et de l'identité est obtenu grâce à l'approche K-Anonymity qui assure la protection des informations telles que l'identité et la localisation, etc. de l'utilisateur[38]
- **L'évaluation des risques** : C'est un élément fondamental de la sécurité IoT qui permet de détecter les nouvelles menaces pesant sur le système. Cela pourrait aider à prévenir les atteintes à la sécurité et à déterminer les meilleures stratégies de sécurité. Un exemple en est la méthode d'évaluation dynamique des risques pour l'Internet des objets.[39]

II.1.2.2 Network Layer :

- **Authentification** : Avec un processus d'authentification approprié et un cryptage point à point, il serait possible d'empêcher un accès illégal aux nœuds de capteurs afin de diffuser de fausses informations
- **Routing Security** : Après le processus d'authentification, des algorithmes de routage sont mis en œuvre pour garantir la confidentialité des échanges de données entre les nœuds de capteurs et les systèmes de traitement. Le routage source, dans lesquelles les données à transmettre sont stockées sous la forme de paquets qui sont ensuite envoyés au système de traitement après avoir été analysés par les nœuds intermédiaires. Routage en boucle dans lequel seule l'adresse de la destination des données est connue. La sécurité du routage est assurée par la fourniture de plusieurs chemins pour le routage des données, ce qui améliore la capacité du système à détecter une erreur et à continuer à fonctionner en cas de défaillance du système.[40]
- **La confidentialité des données** : Les mécanismes de contrôle de la sécurité surveillent le système pour détecter tout type d'intrusion. Les méthodes d'intégrité des données sont implémentées pour garantir que les données reçues à l'autre extrémité sont identiques à celles d'origine.

II.1.2.3 Middle-ware Layer :

- **Authentification :** Ceci est exactement similaire à celui du processus d'identification dans l'une ou l'autre des couches, sauf que cette couche encourage les authentifications de certains services coopérants, ce qui signifie que les utilisateurs peuvent même choisir les informations associées à partager avec les services.[41]

- **Détection d'intrusion :** Génération d'une alarme en cas d'activité suspecte dans le système en raison de la surveillance continue et en tenant un journal des activités de l'intrus qui pourrait aider à le localiser. Il existe différentes techniques de détection d'intrusion existantes, notamment l'approche de l'exploration de données et la détection d'anomalies.[36]
- **L'évaluation des risques :** L'évaluation des risques justifie les stratégies de sécurité efficaces et apporte des améliorations à la structure de sécurité existante.[36]
- **Sécurité des données :** La sécurité des données est assurée par diverses technologies de cryptage qui empêchent les menaces de vol de données. De plus, pour empêcher d'autres utilisateurs malveillants de créer des activités malveillantes, des pare-feu AntiDos et des logiciels espions et malwares à jour sont également introduits.[36]

II.1.3 Les défis de sécurité

Si l'IoT développe notre vie, nous apporte plus de confort et nous aide à surmonter nos différents problèmes, il comporte aussi des risques qui touchent la confidentialité de nos données personnelles. Nous allons citer quelques-unes :

II.1.3.1 Perception Layer :

- **Accès non autorisé aux balises.** En raison du manque de bon mécanisme d'authentification dans un grand nombre de systèmes RFID, les tags peuvent être consultés par quelqu'un sans autorisation. L'attaquant ne peut pas simplement lire les données, mais celles-ci peuvent également être modifiées, voire supprimées.[42]
- **Clonage de balises**. Créer une réplique du tag et de le compromettre de manière que le lecteur ne puisse pas.[43]
- **Eavesdropping**. En raison des caractéristiques sans fil de la RFID, il devient très facile pour l'attaquant de détecter les informations confidentielles telles que les mots de passe ou toute autre donnée circulant d'un lecteur à l'autre ou d'un lecteur à l'autre, ce qui le rend vulnérable car il peut le faire. Utiliser de manière méprisable.[44]
- **Spoofing**. C'est ce qui se passe lorsqu'un attaquant diffuse de fausses informations sur les systèmes RFID et lui fait assumer faussement son originalité, ce qui les fait apparaître à partir de la source originale.[45]
- **RF Jamming**. Les étiquettes RFID peuvent également être compromises par une sorte d'attaque DoS dans laquelle la communication via des signaux RF est interrompue par un excès de signaux de bruit.[46]

- **Capture de nœuds:** Les nœuds clés sont facilement contrôlés par les attaquants tels que les nœuds de passerelle. Toutes les informations, y compris la clé de communication du groupe, la clé radio, la clé correspondante, etc., peuvent fuir, puis menacer la sécurité de l'ensemble du réseau.[47]
- **Timing Attack:** en analysant le temps nécessaire à l'exécution de l'algorithme de chiffrement, pour obtenir des informations de clé.[47]
- **Menaces liées au routage:** l'attaquant peut créer des boucles de routage, causer ou empêcher la transmission sur le réseau, étendre ou raccourcir le chemin source, former des messages d'erreur, augmenter le délai de bout en bout, etc.[47]
- **Replay Attack:** L'attaquant envoie un paquet reçu par l'hôte de destination, afin d'obtenir la confiance du système. Il n'est utilisé que dans le traitement de l'authentification et détruit la validité de la certification.[47]
- **SCA (attaque sur le canal latéral):** L'attaquant attaque les périphériques de chiffrement par le biais des informations de fuite du canal latéral au cours du processus de fonctionnement du périphérique, telles que la consommation de temps, la consommation électrique ou le rayonnement électromagnétique.[47]

II.1.3.2 Network Layer :

Wirless sensor network transmit les données à partir des sensors à leurs destinations.

- **Sleep Deprivation Attack.** La privation de sommeil est le type d'attaque qui maintient les nœuds éveillés, ce qui entraîne une consommation accrue de la batterie. Par conséquent, la durée de vie de la batterie est réduite, ce qui entraîne l'arrêt des nœuds[48]
- **Selective forwarding attacks :** les nœuds malveillants refusent de transmettre certains paquets, afin de détruire les chemins de routage du réseau.[49]

Comme la figure suivante l'illustre :

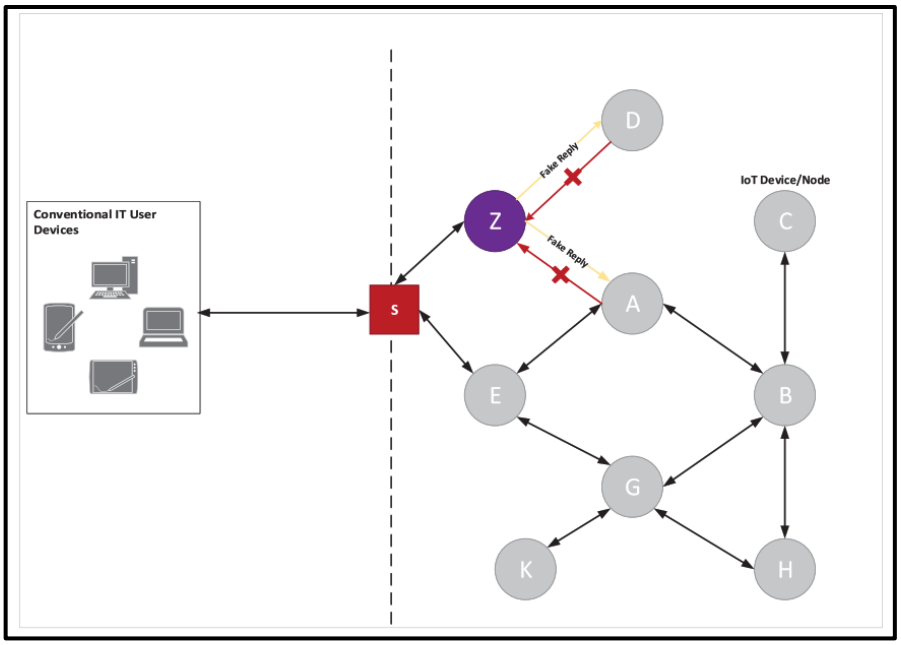

Figure 6 : Selective forwarding attacks

- Les nœuds malveillants refusent de transmettre certains paquets, afin de détruire les chemins de routage du réseau.
- Nœud Z supprime de manière arbitraire les paquets provenant des nœuds A et D.
- Un cas typique est l'attaque de trou noir, dans laquelle le nœud malveillant rejette chaque paquet et n'en transmet aucun. Un autre type est l'attaque par négligence et par cupidité, dans laquelle l'attaquant abandonne certains paquets ou segments d'entre eux. Ces attaques visent à détruire la disponibilité des informations et des services.

- **Sinkhole attacks :** les nœuds malveillants ont pour fonction de diriger le trafic réseau vers un nœud spécifique. Habituellement, ils annoncent un itinéraire particulier du réseau et invitent les autres nœuds à utiliser cet itinéraire.[50]

  Comme la figure suivante l'illustre :

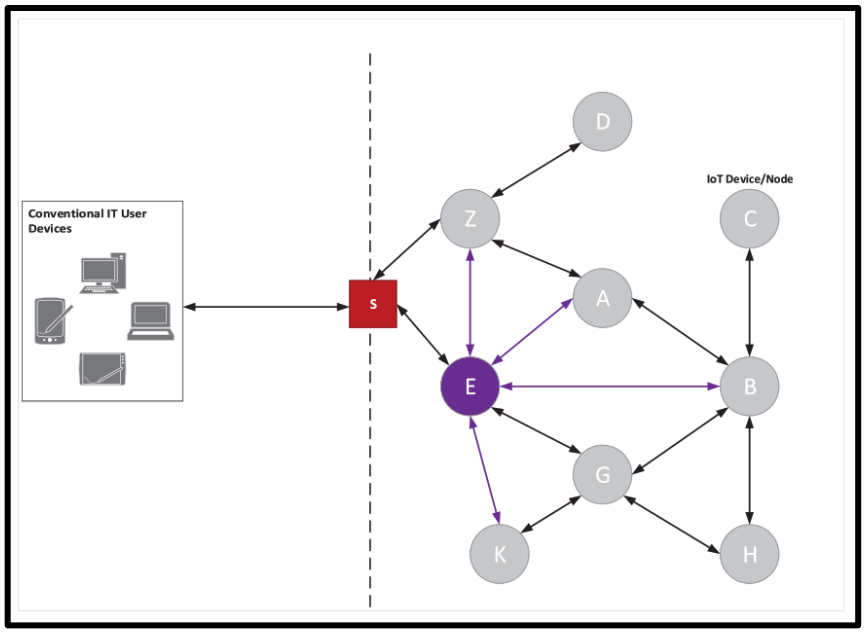

Figure 7 : Sinkhole attacks

- Les nœuds malveillants ont pour fonction de diriger le trafic réseau vers un nœud spécifique.
- Le nœud E s'annonce lui-même. Les nœuds A, B, K et Z sont affectés par l'offensive et transmettent le trafic réseau au nœud E.

- **Wormhole attacks :** les nœuds malveillants créent un lien de communication direct qui est utilisé pour transmettre les données de trafic réseau en ignorant les nœuds intermédiaires.

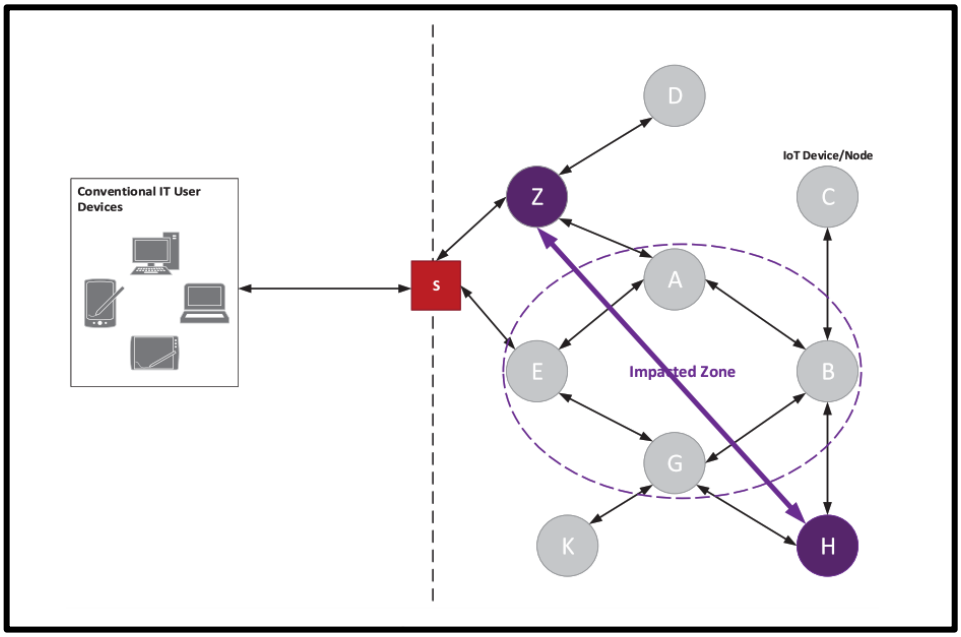

Figure 8 : Wormhole attacks

- Les nœuds malveillants créent un lien de communication direct qui est utilisé pour transférer les données de trafic réseau en ignorant les nœuds intermédiaires.
- Les nœuds H et Z sont connectés via un lien Wormhole.

- **Sybil attacks :** les nœuds malveillants forgent ou créent plusieurs identités afin d'induire en erreur les autres nœuds.[51]

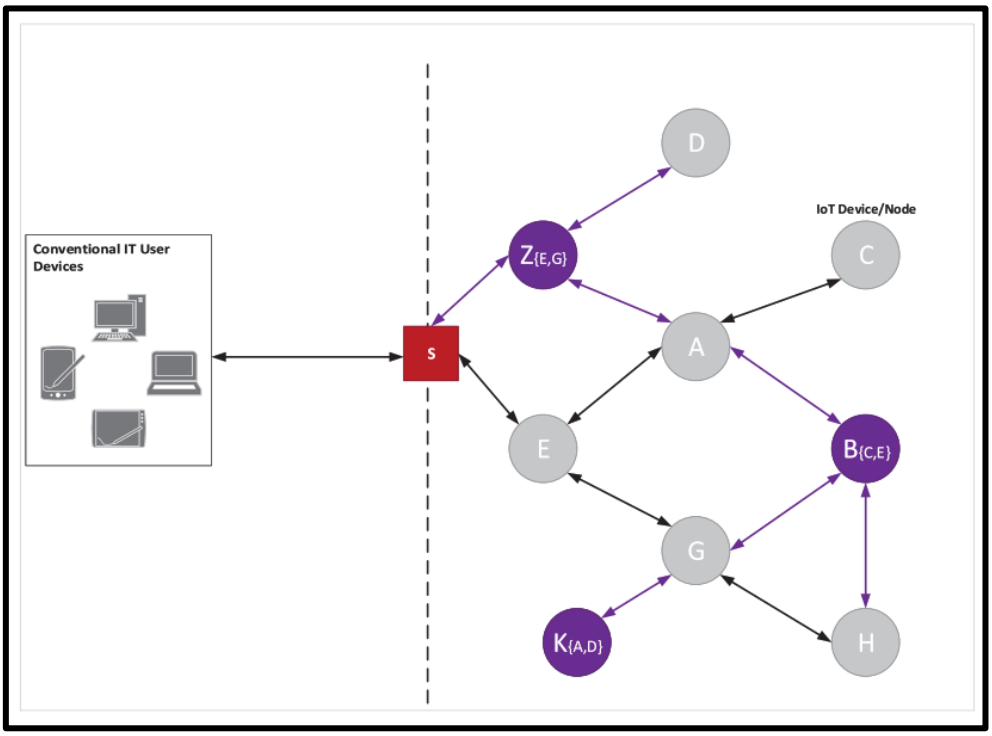

Figure 9 : Sybil attacks

- Les nœuds malveillants forgent ou créent plusieurs identités afin d'induire en erreur les autres nœuds.
- Le but de ces attaques est l'authenticité et la disponibilité des systèmes et des services.

II.1.3.3 Middle-ware Layer :

Cette couche est composée de technologies de stockage, elle est menacée par :

- **L'accès non autorisé**. La couche middle-ware fournit différentes interfaces pour les applications et les installations de stockage de données. L'attaquant peut facilement causer des dommages au système en interdisant l'accès aux services connexes de l'IoT ou en supprimant les données existantes. Un accès non autorisé pourrait donc être fatal pour le système.[36]
- **Ddos Attack :** une attaque ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser.

- **Malicious Insider**. Ce type d'attaque se produit lorsqu'une personne de l'intérieur altère les données pour des avantages personnels ou ceux d'une tierce partie. Les données peuvent être facilement extraites puis modifiées intentionnellement de l'intérieur.[36]

La sécurité est un enjeu majeur de l'IoT pour la protection des données. Nous avons présenté dans ce chapitre les principales problématiques de la sécurité de l'IoT, les techniques utilisées jusqu'à présent ainsi que les différents défis.

Chapitre III: Applications

Dans ce chapitre nous allons voir la configuration du serveur Dell PowerEdge R440, comparaison de ses performances avec d'autres ordinateurs, aussi une comparaison de performances des librairies de python pour le traitement Big Data, Ainsi qu'une application IoT pour la mesure de température.

III.1 Configuration du serveur Dell PowerEdge R440

Le laboratoire LSSC s'est doté d'un serveur Dell PowerEdge R44O dont le but de l'utiliser pour des applications IoT, minimiser le temps des calculs scientifiques et l'exécution des simulations pour améliorer les travaux de recherche.

Le PowerEdge R440 est un serveur qui offre la combinaison de performances et de densité pour le HPC et les déploiements de technologies Web avec un ensemble de fonctions adapté aux environnements d'infrastructure scale-out.

III.1.1 Caractéristiques du serveur

Fonctionnalités	Caractéristiques techniques
Processeur	Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz 8coeurs
Mémoire	16 GB
Stockage	500 GB
Baies de disques	Baies de disques avant : jusqu'à 10 disques SAS/SATA (disque dur/SSD) 2,5 pouces avec jusqu'à 4 disques SSD NVMe 48 To max. Ou jusqu'à 4 disques durs SAS/SATA de 3,5 pouces 56 To max. DVD-ROM, DVD+ RW
Blocs d'alimentation Ventilateurs	450 W or, 550 W platine Six ventilateurs installables à chaud pour une redondance N+1
Dimensionnement	Encombrement : Rack (1U) Châssis : 728,23 mm du panneau avant à la poignée arrière du bloc d'alimentation 714,58 mm de la plaque avant (sans panneau) à la poignée arrière du bloc d'alimentation
Intégré/sur le serveur	iDRAC9 avec Lifecycle Controller Fonction iDRAC Direct iDRAC REST API avec Redfish Module BLE/sans fil Quick Sync 2
Intégrations	Intégrations OpenManage : Microsoft® System Center, VMware® vCenter™, BMC Software
Outils	Gestionnaire de référentiel Dell EMC (Dell EMC Repository Manager)

	Package de mise à jour de Dell EMC Mise à jour système Dell EMC Dell EMC Server Update Utility iDRAC Service Module
Sécurité	TPM 1.2/2.0 (en option) Microprogrammes signés sous forme chiffrée Racine de confiance au niveau de la puce Démarrage sécurisé Verrouillage du système Suppression du système
E/S et ports	Ports avant : vidéo, 1 x USB 2.0, port USB 3.0 disponible, port USB IDRAC Direct dédié Ports arrière : Vidéo, série, 2 ports USB 3.0, port réseau iDRAC dédié Les options de montage également disponibles incluent 2 x HH/HL ou 1 x FH/HL
Systèmes d'exploitation pris en charge	Canonical® Ubuntu® LTS, Citrix® XenServer®, Microsoft Windows Server® avec Hyper-V, Red Hat® Enterprise Linux, SUSE® Linux Enterprise Server, VMware® ESXi

Tableau 1 : Caractéristiques du serveur Dell PowerEdge R440

III.1.2 Configuration

III.1.2.1 Installation de Life Cycle Controller

Le Life Cycle Controller est le système de base des serveurs Dell, il permet la configuration des paramètres matériels et du BIOS, le déploiement de systèmes d'exploitation, la modification des paramètres RAID et l'enregistrement des profils matériels du serveur. Le Lifecycle Controller simplifie la gestion du cycle de vie du serveur : depuis le provisioning, le déploiement, les correctifs et les mises à jour jusqu'à l'entretien et la personnalisation par l'utilisateur, à la fois localement et à distance.

Il est important pendant l'installation de Life Cycle Controller de définir une adresse IP statique pour faciliter l'accès à distance au panel du serveur.

III.1.2.2 Installation de VmWare ESXI 6.5

Selon les exigences des enseignants chercheurs du laboratoire, chacun doit avoir un accès au serveur à distance et la possibilité de lancer ses travaux en conservant la confidentialité. C'est pour cela que le système VmWare ESXI représente la meilleure solution, car il donne la possibilité de créer plusieurs machines virtuelles et attribué des droits spécifiques à chaque compte crée sur une machine ou plusieurs.

Donc chaque enseignant a son propre compte et le droit d'accéder à sa propre machine virtuelle sans pouvoir voir les autres machines.

III.2 Performance du serveur DELL PowerEdge R440

Dans nos jours, des volumes massifs de données sont collectés à chaque secondes, et demandent du temps pour leurs traitements. C'est pour cela qu'on recourt à des solutions cloud pour l'analyse Big Data, car les données sont traitées dans un laps de temps très court, voire en temps réel.

Pour que La faculté des sciences et techniques de Fès dispose des bonnes informations et puisse décider en temps utile, le laboratoire LSIA a décidé d'installer un serveur DELL PowerEdge R440.

L'analyse Big Data requiert également de puissants algorithmes et de bonnes configurations matériels afin de rendre très rapidement compréhensibles toutes ces données et de les exploiter de manière efficace dans un environnement qui évolue sans cesse.

Pour atteindre ce but, nous avons comparé trois algorithmes de traitement des données : PANDAS – NUMPY – DASK, afin de déterminer lequel est le plus performant, ainsi qu'on a effectué ces tests sur deux autres machines pour comparer les performances du serveur installé.

III.2.1 Configuration du matériel

	Processeur	Cores	Ram	Type Disque dur
Dell R440	Xeon silver 4110 2.10 GHz	4	16 go	HDD
Dell R440	Xeon silver 4110 2.10 GHz	8	16 go	HDD
Macbook Pro	I7-4850HQ 2.30GHz	8	16 go	SSD
Dell Latitude E5470	I5-6440HQ 2.60GHz	4	8 go	SDD

Tableau 2 : Configuration des machines utilisées pour les test de performances

III.2.2 Librairies Python utilisées

III.2.2.1 NumPy

NumPy signifie Numeric Python, NumPy est le paquet fondamental pour l'informatique scientifique avec Python. Il est utilisé pour effectuer des opérations numériques sur les tableaux. NumPy est mieux que la liste python en termes de taille, de vitesse et de fonctionnalité.

- **Installation de NumPy**

Si on a Anaconda, on peut simplement installer NumPy depuis notre terminal ou l'invite de commande en utilisant :

```
conda install numpy
```

Si on n'a pas Anaconda sur notre ordinateur, on installe NumPy depuis notre terminal en utilisant :

```
pip install numpy
```

Numpy peut être importé dans le notebook en utilisant :

```
import numpy as np
```

- **Les tableaux**

L'objet principal de NumPy est le tableau multidimensionnel homogène. Il s'agit d'une table avec les mêmes éléments de type, c'est-à-dire des entiers ou des caractères (homogènes), généralement des entiers. Dans NumPy, les dimensions sont appelées axes. Le nombre d'axes est appelé le rang.

Il y a plusieurs façons de créer un tableau dans NumPy comme np.array, np.zeros, no.ones, etc. Chacun d'entre eux offre une certaine flexibilité.

Commande	exemple

np.array	>>> a = np.array([1, 2, 3])
	>>> type(a)
	<type 'numpy.ndarray'>
	>>> b = np.array((3, 4, 5))
	>>> type(b)
	<type 'numpy.ndarray'>

np.ones	>>> np.ones((3,4), dtype=np.int16) array([[1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]])
np.full	>>> np.full((3,4), 0.11) array([[0.11, 0.11, 0.11, 0.11],

	[0.11, 0.11, 0.11, 0.11], [0.11, 0.11, 0.11, 0.11]])

np.arange	>>> np.arange(10, 30, 5)
	array([10, 15, 20, 25])
	
	>>> np.arange(0, 2, 0.3)
	array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8])

np.linspace	>>> np.linspace(0, 5/3, 6) array([0. , 0.33333333 , 0.66666667 , 1. , 1.33333333 1.66666667])
np.random.rand(2,3)	>>> np.random.rand(2,3) array([[0.55365951, 0.60150511, 0.36113117], [0.5388662 , 0.06929014, 0.07908068]])
np.empty((2,3))	>>> np.empty((2,3)) array([[0.21288689, 0.20662218, 0.78018623], [0.35294004, 0.07347101, 0.54552084]])

Tableau 3 : Exemple des commandes NumPy

Certains attributs importants d'un objet NumPy sont:

- **Ndim** : affiche la dimension du tableau
- **Shape** : retourne un tuple d'entiers indiquant la taille du tableau
- **Size** : renvoie le nombre total d'éléments dans le tableau NumPy
- **Dtype** : renvoie le type d'éléments dans le tableau, c.-à-d., Int64, caractère
- **Itemsize** : retourne la taille en octets de chaque élément
- **Reshape** : remodèle le tableau NumPy

Les éléments de tableau NumPy sont accessibles par indexation. Voici quelques exemples utiles :

- **A[2:5]** imprimera les éléments 2 à 4. L'index dans les tableaux NumPy commence à 0
- **A[2::2]** imprimera les éléments 2 pour finir de sauter 2 éléments
- **A[::-1]** imprimera le tableau dans l'ordre inverse
- **A[1:]** s'imprimera de la ligne 1 à la fin

Les tableaux NumPy sont capables d'effectuer toutes les opérations de base telles que l'addition, la soustraction, le produit dans le sens des éléments, le produit à points matriciels, la division dans le sens des éléments, le modulo dans le sens des éléments, les exposants dans le sens des éléments et les opérations conditionnelles.

Une caractéristique importante des tableaux NumPy est le broadcasting.

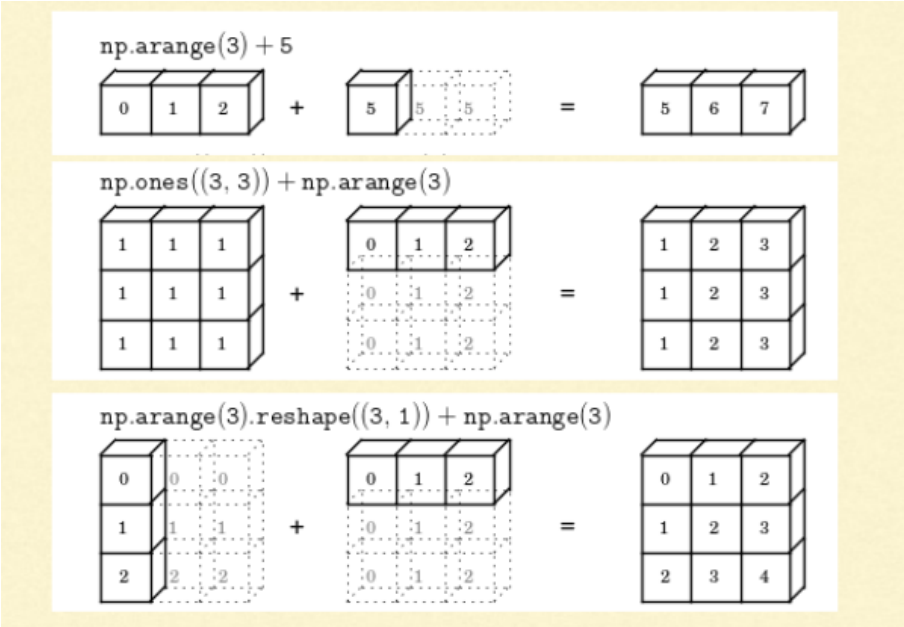

Figure 10 : Tableau NumPy

En général, quand NumPy s'attend à des tableaux de la même forme mais constate que ce n'est pas le cas, il applique les règles dites de broadcasting.

Fondamentalement, il y a deux règles de broadcasting à retenir:

1. Pour les tableaux qui n'ont pas le même rang, alors un 1 sera pré-terminé pour les tableaux de classement plus petits jusqu'à ce que leurs rangs correspondent. Par exemple, lors de l'ajout des tableaux A et B de tailles (3,3) et (,3) [rang 2 et rang 1], 1 sera prédéfini à la dimension du tableau B pour le rendre (1,3) [rang = 2]. Les deux ensembles sont compatibles lorsque leurs dimensions sont égales ou que l'une des dimensions est 1.
2. Lorsque l'une ou l'autre des dimensions comparées est l'une, l'autre est utilisée, c'est-à-dire que les dimensions de la taille 1 sont étirées ou « copiées » pour correspondre à

l'autre. Par exemple, en ajoutant un tableau 2D A de forme (3,3) à un ndarray 2D de forme B (1, 3). NumPy appliquera la règle ci-dessus de diffusion. Il doit étirer le tableau B et reproduire la première ligne 3 fois pour faire le tableau B des dimensions (3,3) et effectuer l'opération.

NumPy fournit des fonctions mathématiques et statistiques de base comme moyenne, min, max, sum, prod, std, var, sommation à travers différents axes, transposition d'une matrice, etc.

Une caractéristique particulière de NumPy d'intérêt est de résoudre un système d'équations linéaires. NumPy a des fonctions pour résoudre des équations linéaires. Par exemple,

2x + 6y = 6

5x + 3y = -9

Peut être résolu par NumPy :

```
>>> coeffs  = np.array([[2, 6], [5, 3]])
>>> depvars = np.array([6, -9])
>>> solution = linalg.solve(coeffs, depvars)
>>> solution
array([-3.,  2.])
```

III.2.2.2 Pandas

Similaire à NumPy, Pandas est l'une des bibliothèques python les plus largement utilisées dans la science des données. Il offre des structures et des outils d'analyse de données performants et faciles à utiliser. Contrairement à la bibliothèque NumPy qui fournit des objets pour les tableaux multidimensionnels, Pandas fournit en mémoire 2d objet de table appelé dataframe. C'est comme un chiffrier avec les noms de colonne et les étiquettes de ligne.

Par conséquent, avec les tables 2d, pandas est capable de fournir de nombreuses fonctionnalités supplémentaires telles que la création de tables pivots, le calcul de colonnes basées sur d'autres colonnes et des graphiques de tracé. Pandas peut être importé en Python en utilisant :

```
import pandas as pd
```

Voici quelques structures de données couramment utilisées dans les pandas :

- **Series objects**: tableau 1D, semblable à une colonne dans un chiffrier.
- **Dataframe objects** : tableau 2D, similaire à un tableur.
- **Panel objects** : Dictionnaire des dataframes, similaire à la feuille dans MS Excel.

L'objet Pandas Series est créé en utilisant la fonction pd. Series. Chaque ligne est fournie avec un index et par défaut est attribué des valeurs numériques à partir de 0. Comme NumPy, Pandas fournissent également les fonctionnalités mathématiques de base comme l'addition, la soustraction et les opérations conditionnelles et le broadcasting.

L'objet Pandas dataframe représente une feuille de calcul avec les valeurs des cellules, les noms des colonnes et les étiquettes d'index des lignes. Dataframe peut être visualisé comme dictionnaire de série. Les lignes et colonnes de dataframe sont simples et intuitives à accéder. Pandas fournit également des fonctionnalités de type SQL pour filtrer, trier les lignes en fonction des conditions. Par exemple :

```
>>> people_dict = { "weight": pd.Series([68, 83, 112],index=["alice",
"bob", "charles"]),     "birthyear": pd.Series([1984, 1985, 1992],
index=["bob", "alice", "charles"], name="year"),

    "children": pd.Series([0, 3], index=["charles", "bob"]),

    "hobby":   pd.Series(["Biking",   "Dancing"],   index=["alice",
"bob"]),}

>>> people = pd.DataFrame(people_dict)
>>> people
```

	birthyear	children	hobby	weight
alice	1985	NaN	Biking	68
bob	1984	3.0	Dancing	83
charles	1992	0.0	NaN	112

Figure 11 : Résultat d'affichage

```
>>> people[people["birthyear"] < 1990]
```

	birthyear	children	hobby	weight
alice	1985	NaN	Biking	68
bob	1984	3.0	Dancing	83

Figure 12 : Résultat d'affichage

De nouvelles colonnes et lignes peuvent facilement être ajoutées à la base de données. En plus des fonctionnalités de base, la base de données pandas peut être triée par colonne particulière.

Les bases de données peuvent également être facilement exportées et importées de CSV, Excel, JSON, HTML et SQL. Voici d'autres méthodes essentielles qui sont présentes dans les bases de données :

- **head()** : renvoie les 5 premières lignes de l'objet dataframe
- **tail()** : renvoie les 5 dernières lignes de la base de données
- **info()** : affiche le résumé de la base de données
- **describe()** : donne un bon aperçu des principales valeurs agrégées sur chaque colonne

III.2.2.3 Dask

Dask est une bibliothèque de calcul parallèle qui n'aide pas seulement à paralléliser les outils d'apprentissage automatique existants (Pandas et Numpy), mais permet également de paralléliser les tâches/fonctions de bas niveau et peut gérer des interactions complexes entre ces fonctions en faisant un graphique des tâches. Ceci est similaire aux modules de Threading ou multiprocessing de Python.

Dask parallélise les tâches qui lui sont confiées en faisant un graphique des interactions entre les tâches. Il sera vraiment utile de visualiser ce que nous faisons en utilisant Dask.Visualize() méthode qui est disponible avec tous ses types de données et avec la chaîne complexe de tâches que nous calculons. Cette méthode va produire un graphique de nos tâches, et si nos tâches ont de nombreux nœuds à chaque niveau (c.-à-d. notre structure de chaîne de tâches ont de nombreuses tâches indépendantes à de nombreux niveaux, comme tâche parallélisable sur des morceaux de données), puis Dask sera en mesure de les paralléliser.

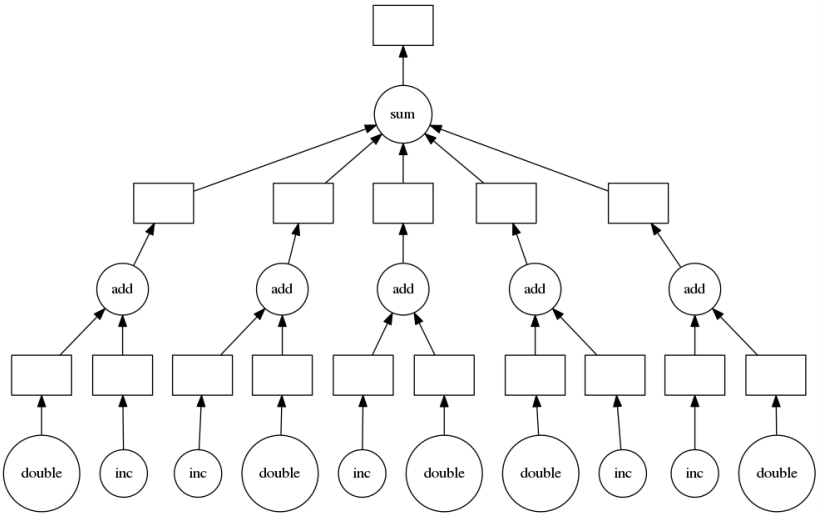

Figure 13 : Graphique des interactions entre les tâches Dask

Types des données

Chaque type de données dans Dask fournit une version distribuée des types de données existants, tels que dataframe de Pandas, ndarray de numpy, et la liste de Python. Ces types de données peuvent être plus grands que notre mémoire, Dask exécutera des calculs sur nos données parallèle(y) de manière bloquée. Bloqués dans le sens où ils effectuent de grands

calculs en effectuant de nombreux petits calculs, c.-à-d. dans les blocs, et le nombre de blocs sont le nombre total de morceaux.

Array : Dask Array fonctionne sur de très grands tableaux, en les divisant en morceaux et en exécutant ces blocs parallèlement.Dask Array peut lire à partir de n'importe quelle structure de tableau comme celle qui supporte numpy comme le tranchage et a la propriété .shape en utilisant la méthode dask.array.from_array. Il peut également lire des fichiers .npy et .zarr.

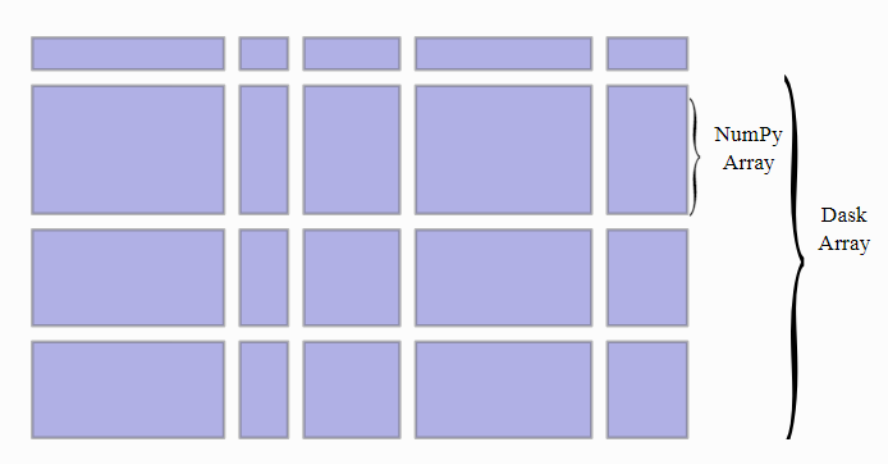

Figure 14 : Dask Array

```
import dask.array as da
import numpy as np
arr = numpy.random.randint(1, 1000, (10000, 10000))
darr = da.from_array(arr, chunks=(1000, 1000))
# darr.npartitioins
# 100
```

Il fera des morceaux, chacun de la taille (1000, 1000)

Dask évalue paresseusement chaque méthode. Donc, pour calculer la valeur d'une fonction, nous devons utiliser la méthode .compute(). Il calculera le résultat en parallèle dans des blocs, parallélisant chaque tâche indépendante à ce moment.

```
result = darr.compute()
```

Dataframe : Semblable aux Dask arrays, les Dask Dataframe permettent de faire des calculs parallèles sur de très grands fichiers de données, qui ne correspondent pas à la mémoire, en divisant les fichiers en blocs et en multipliant les fonctions de calcul en blocs parallèles.

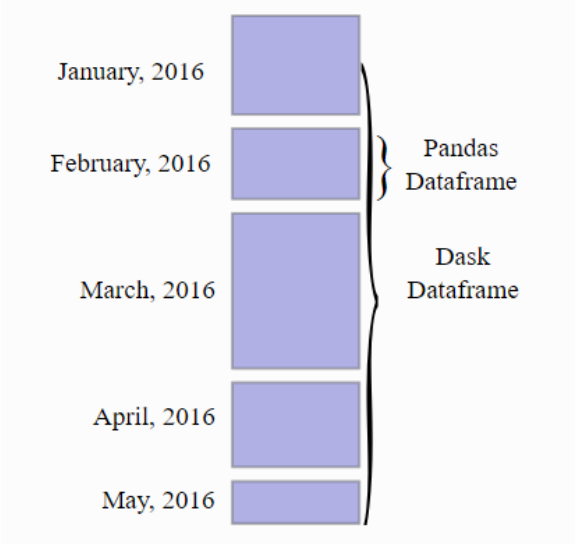

Figure 15 : Dask Dataframe

```
import dask.dataframe as dd

df = dd.read_csv("BigFile(s).csv", blocksize=50e6)
```

Nous pouvons maintenant appliquer/utiliser la plupart des fonctions disponibles dans la bibliothèque pandas et l'appliquer ici.

```
agg = df.groupby(["column"]).aggregate(["sum", "mean", "max", "min"])
```

```
agg.columns = new_column_names # see in notebook

df_new = df.merge(agg.reset_index(), on="column", how="left")

df_new.compute().head()
```

Bag : Dask Bags parallélise le calcul sur la liste de Python comme les objets qui contiennent des éléments de nombreux types de données. Il est utile lorsque nous essayons de traiter des données semi-structurées comme des blobs JSON ou des fichiers journaux.

```
import dask.bag as db

b = db.from_txt("BigSemiStructuredData.txt")

b.take(1)
```

Les Dask Bags lisent ligne par ligne. Dask Bag implémente des opérations comme map, filter, fold, et groupby sur de telles collections d'objets Python. Il le fait en parallèle avec une petite empreinte mémoire en utilisant des itérateurs Python. Il est similaire à une version parallèle de PyToolz ou à une version Pythonique de PySpark RDD.

Delayed

Si notre tâche est un peu simple et que nous ne sommes pas en mesure ou ne voulons pas le faire avec ces collections de haut niveau, alors nous pouvons utiliser des ordonnanceurs de bas niveau qui nous aident à paralléliser notre code/algorithme en utilisant l'interface dask.late. dask.delayed fait aussi des calculs paresseux.

```
import dask.delayed as delay

@delay
def sq(x):
    return x**2
@delay
def add(x, y):
    return x+y
@delay
def sum(arr):
    sum=0
    for i in range(len(arr)): sum+=arr[i]
    return sum
```

III.2.3 Expérimentation et résultats

III.2.3.1 Applications 1

Pour la première application, on a utilisé la base de données Le Bay Area Bike Share. (Le Bay Area Bike Share permet des excursions à vélo rapides, faciles et abordables dans la région de San Francisco Bay.). Cette base de données contient 71millions lignes.

Pour tester les performances de notre serveur Dell PowerEdge R440, on a effectué trois tests, qui sont exécuté aussi sur deux autres ordinateurs portables (Voir configuration Matériel). Pour chaque test on calcule le temps de traitement.

Ceci nous permets aussi de définir quelle librairie python est plus performante pour le traitement des grandes bases de données.

III.2.3.1.1 **Test 1 : Calcul de la proportion des vélos disponibles pour un nombre de lignes.**

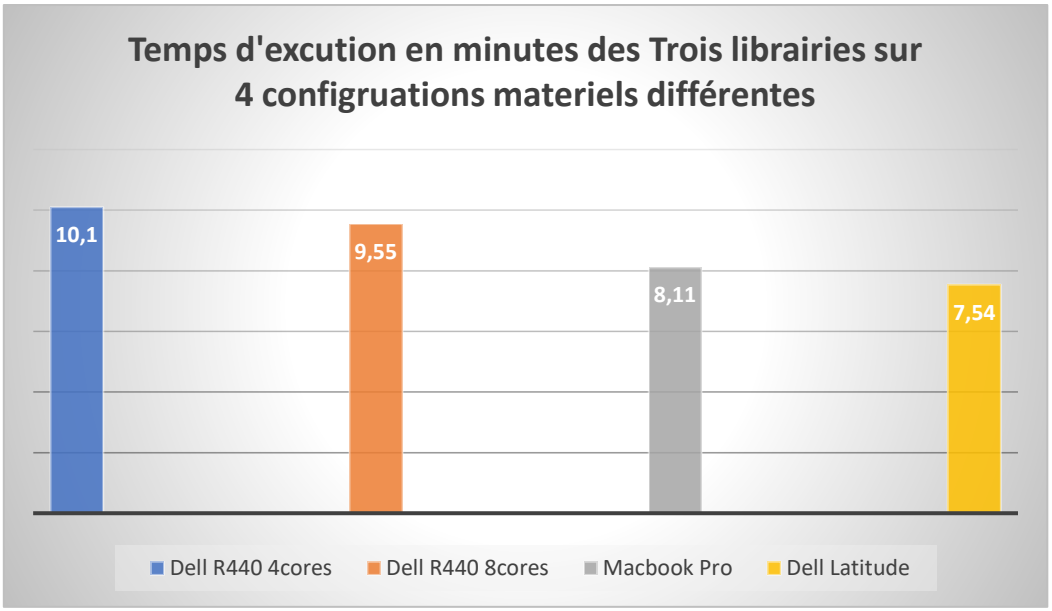

Figure 16 : Test 1 temps d'exécution des 3 librairies sur 4 config matériels différentes

- Performance des librairies en fonction de temps et de nombre des lignes :
    - Dell Latitude :

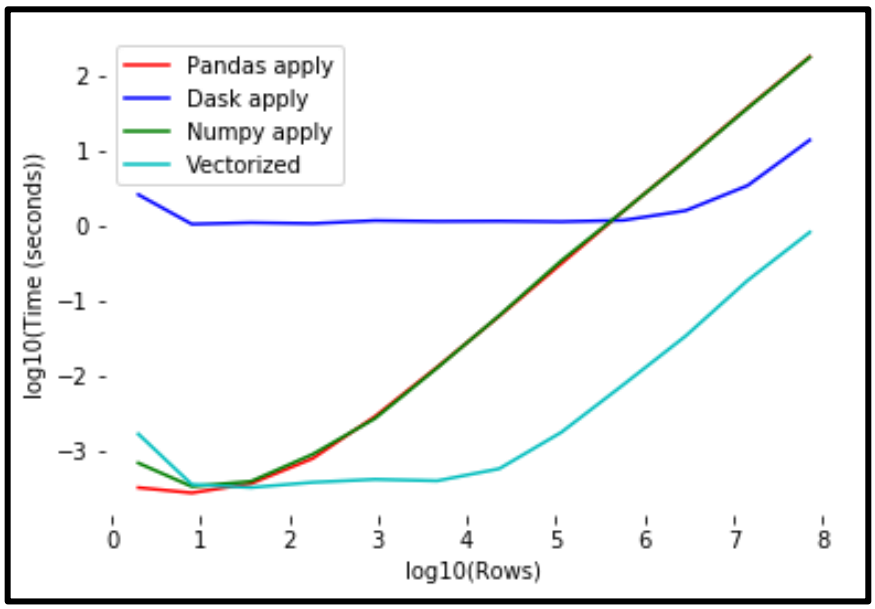

Figure 17 : Test 1 log10 du temps d'exécution des librairies sur Dell Latitude

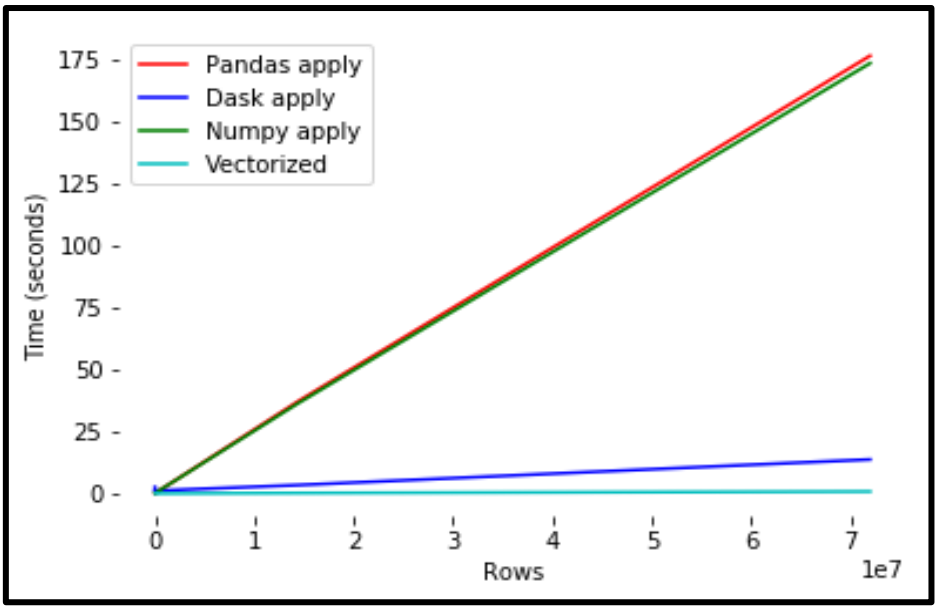

Figure 18 : Test 1 temps d'exécution des librairie sur Dell Latitude

- **Macbook Pro :**

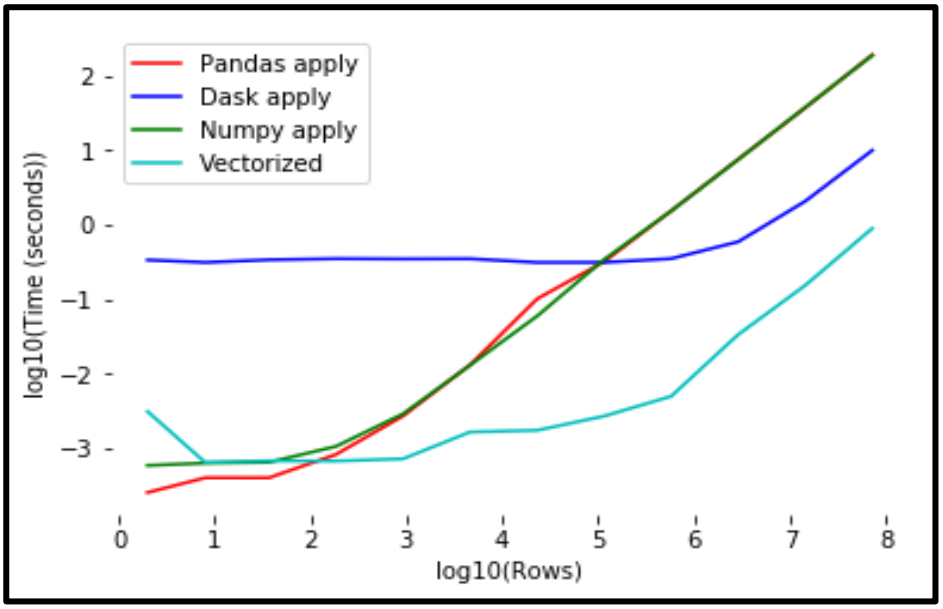

Figure 19 : Test 1 lo10 temps d'exécution des librairies sur MacBook Pro

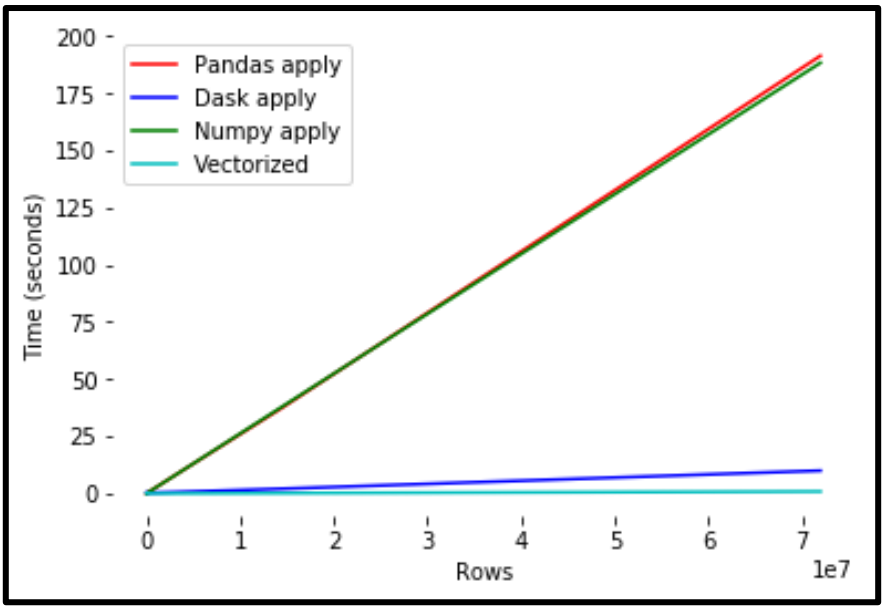

Figure 20 : Test 1 temps d'exécution des librairies sur MacBook Pro

- **Dell R440 4cores :**

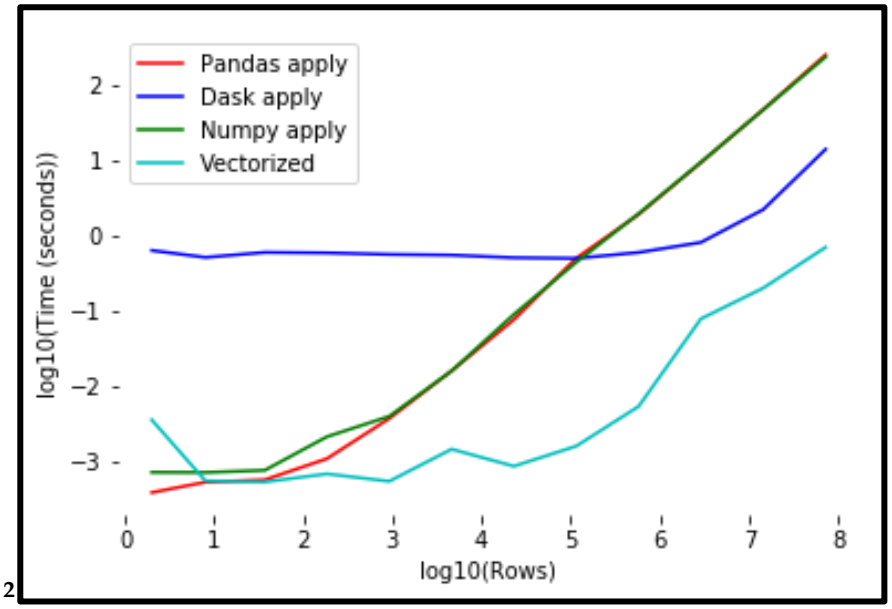

Figure 21 : Test 1 log10 temps d'exécution des librairies sur serveur 4cores

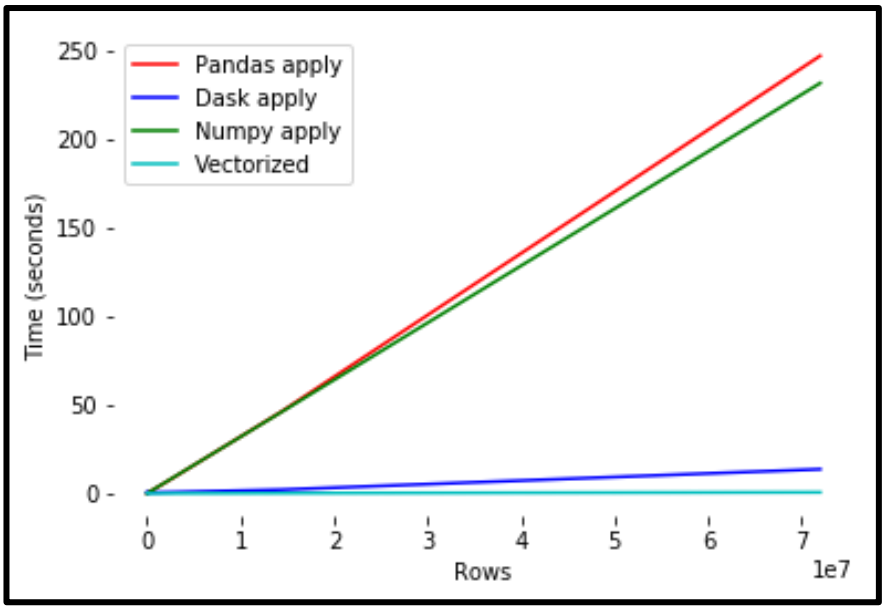

Figure 22 : Test 1 temps d'exécution des librairies sur serveur 4cores

- **Dell R440 8cores :**

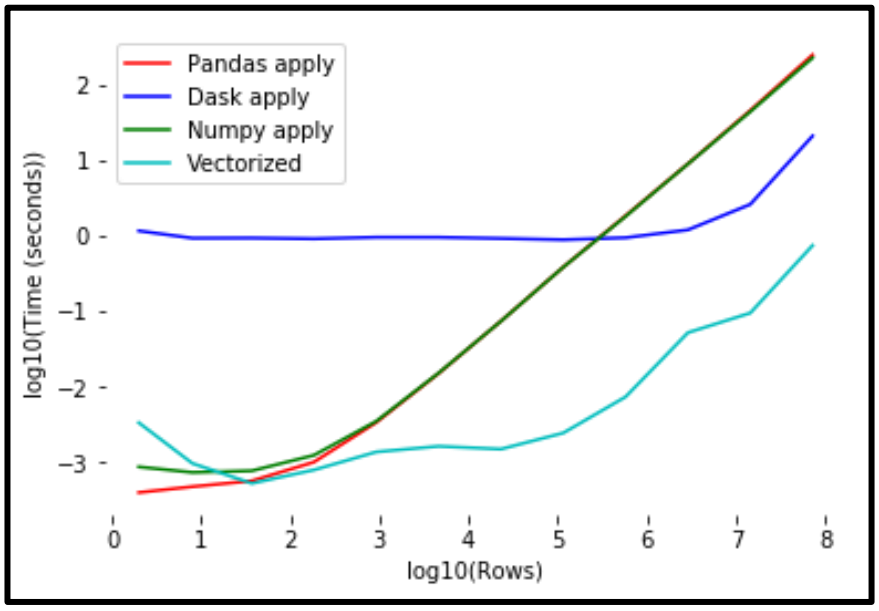

Figure 23 : Test 1 log10 temps d'exécution des librairies sur serveur 8cores

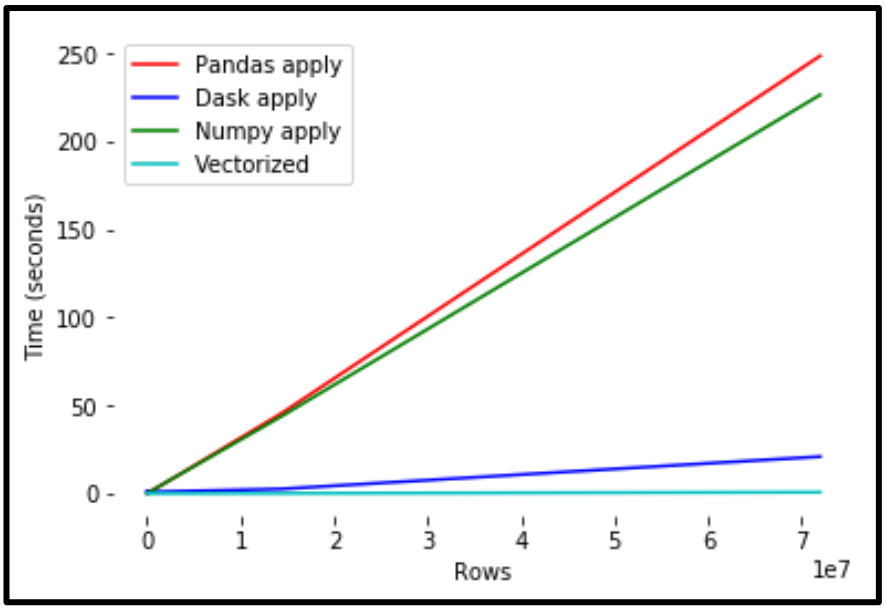

Figure 24 : Test 1 temps d'exécution des librairies sur serveur 8cores

III.2.3.1.2 **Test 2 : Calculer le total du temps de disponibilité des vélos et quais pour un nombre de ligne.**

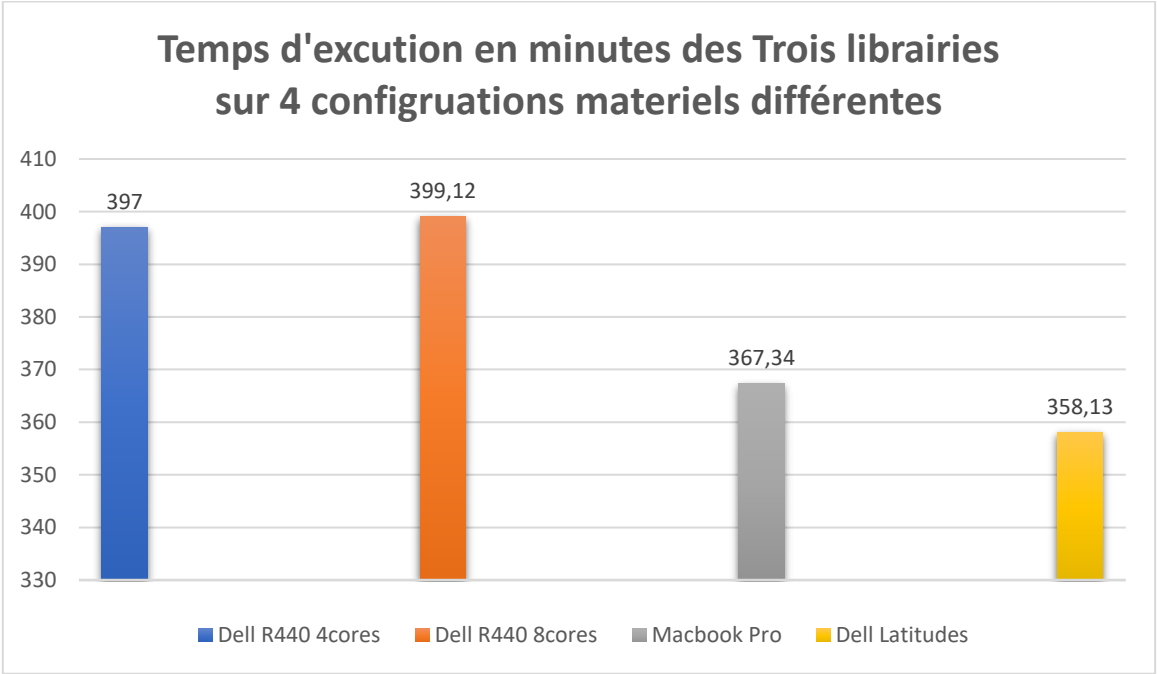

Figure 25 : Test 2 temps d'exécution des librairies sur 4 configurations matériels différentes

- **Performance des librairies en fonction de temps et de nombre des lignes :**
  - **Dell Latitude :**

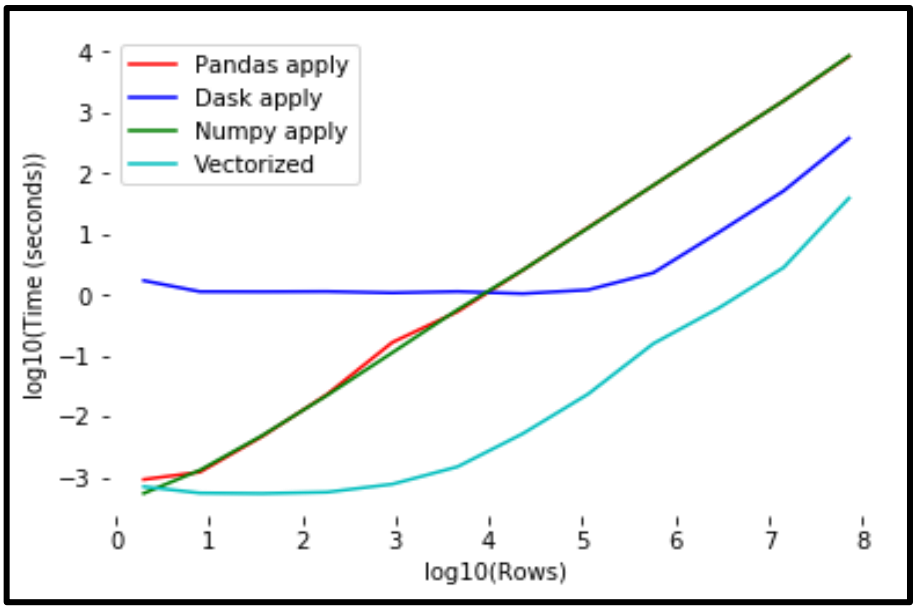

Figure 26 : Test 2 log10 temps d'exécution des librairies sur Dell Latitude

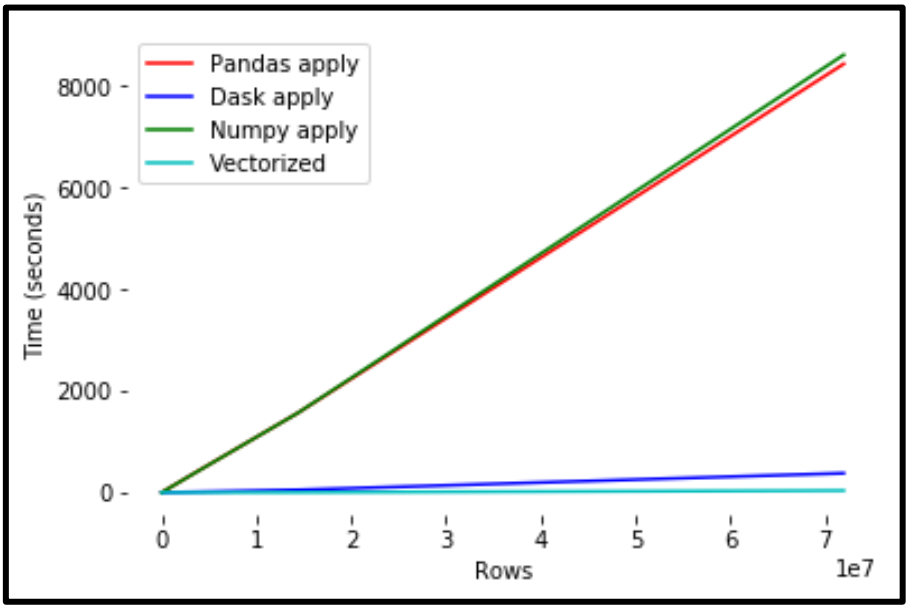

Figure 27 : Test 2 temps d'exécution des librairies sur Dell Latitude

- **Macbook Pro :**

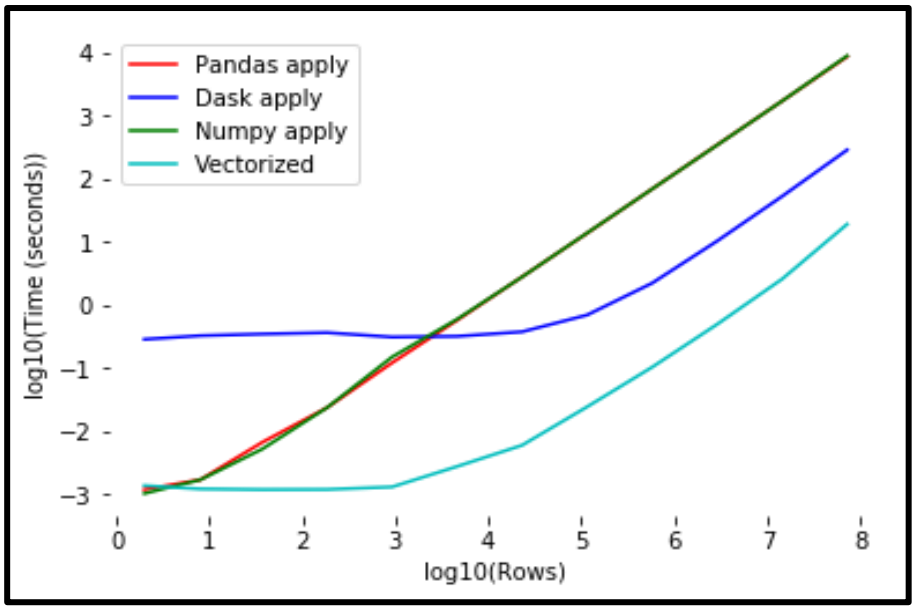

Figure 28 : Test 2 log10 temps d'exécution des librairies MacBook Pro

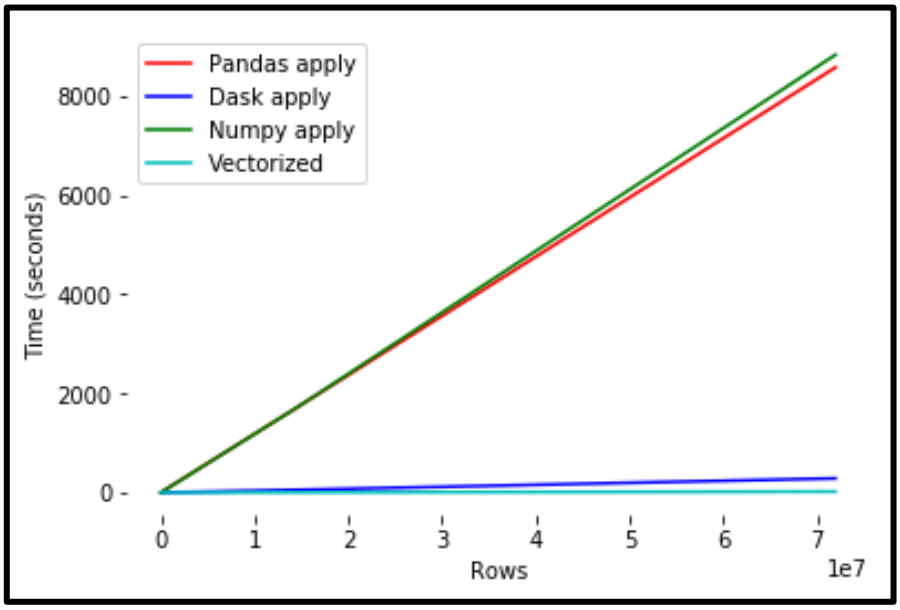

Figure 29 : Test 2 temps d'exécution des librairies MacBook Pro

- **Dell R440 4cores :**

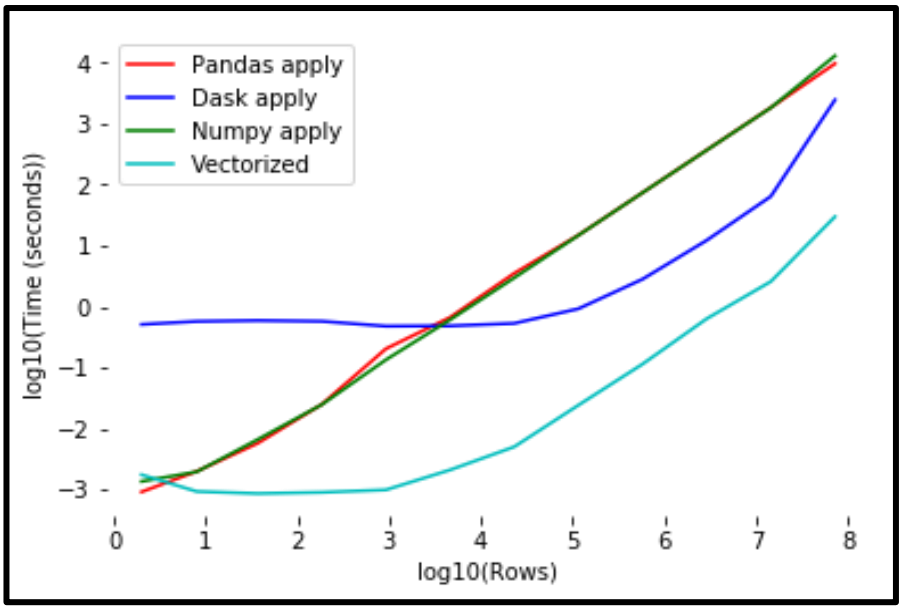

Figure 30 : Test 2 log10 temps d'exécution des librairies

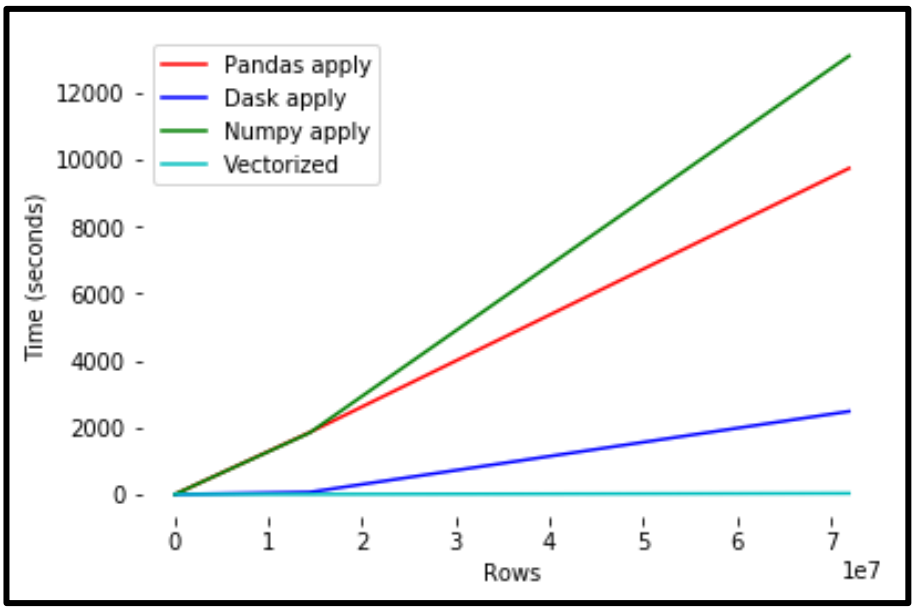

Figure 31 : Test 2 temps d'exécution des librairies sur serveur 4cores

- **Dell R440 8cores :**

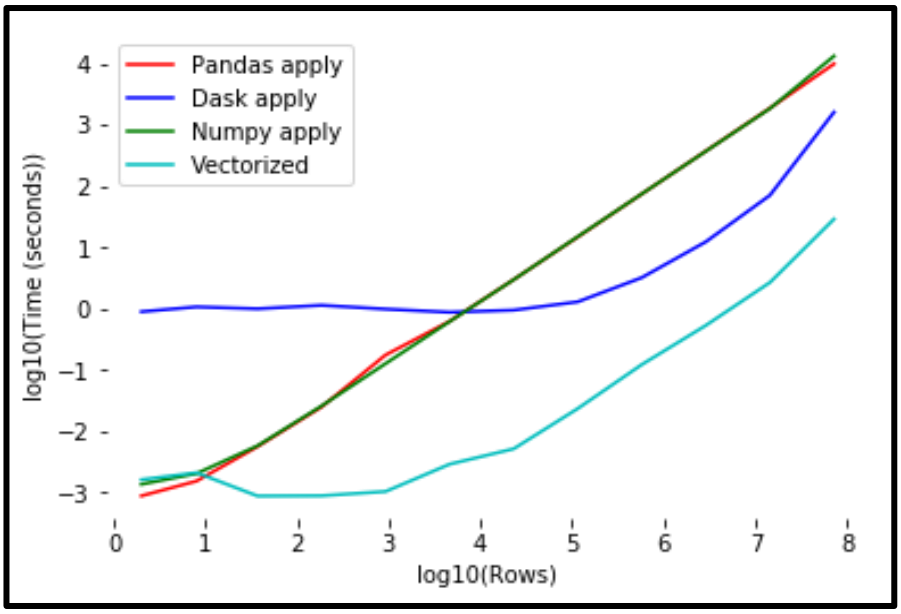

Figure 32 : Test 2 temps d'exécution des librairies sur serveur 8cores

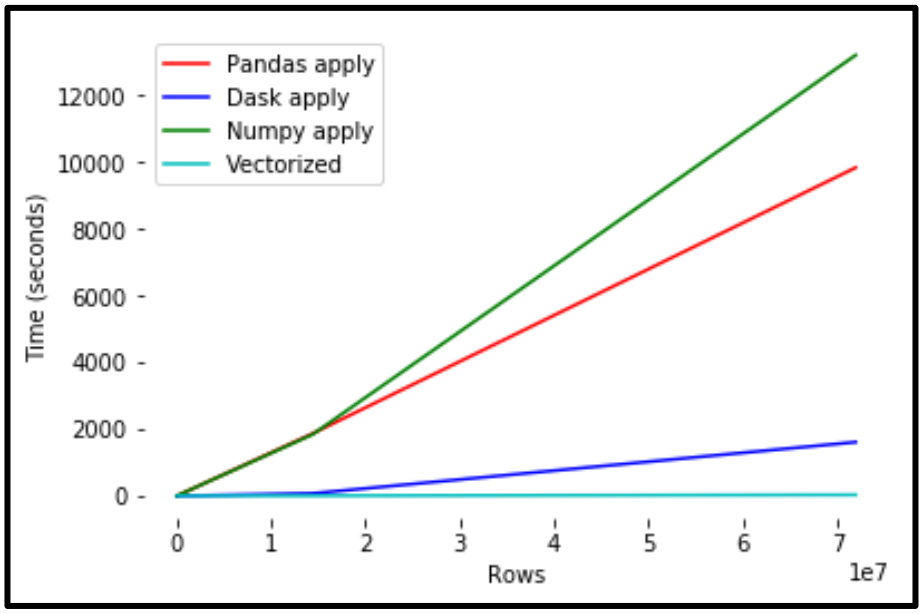

Figure 33 : Test 2 temps d'exécution des librairies sur serveur 8cores

III.2.3.1.3 **Test 3 : Définir si le moment de disponibilité du vélos ou quai est le matin ou non**

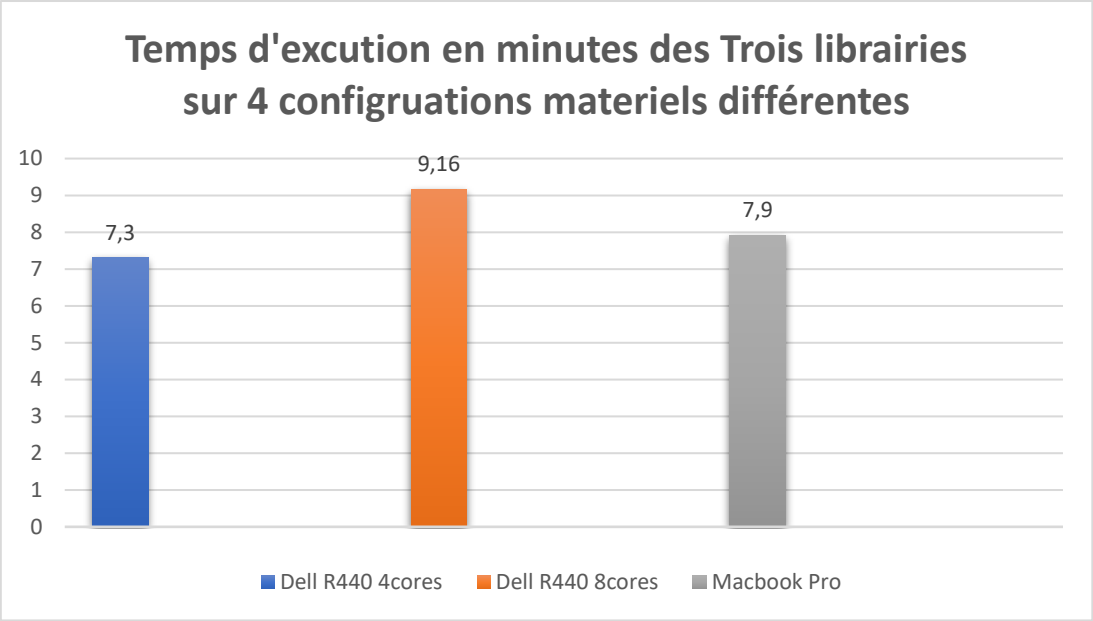

Figure 34 : Test 3 temps d'exécution des librairies sur 4 configurations matériels différentes

L'ordinateur Dell Latitude E5470 n'a pas pu terminer ce test à cause d'une insuffisance de la mémoire.

- **Performance des librairies en fonction de temps et de nombre des lignes :**
  - **Macbook Pro :**

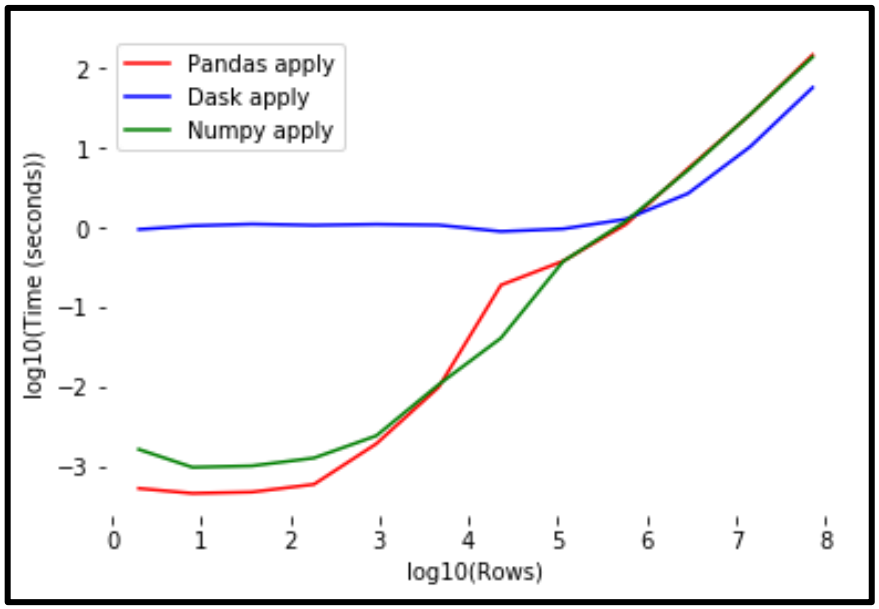

Figure 35 : Test 3 log10 temps d'exécution des librairies sur MacBook Pro

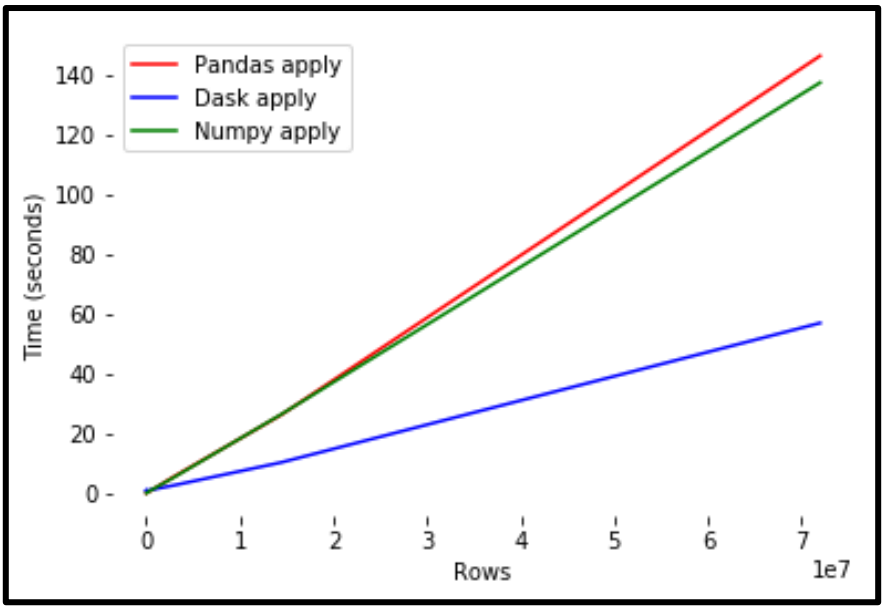

Figure 36 : Test 3 temps d'exécution des librairies sur MacBook Pro

- **Dell R440 4cores :**

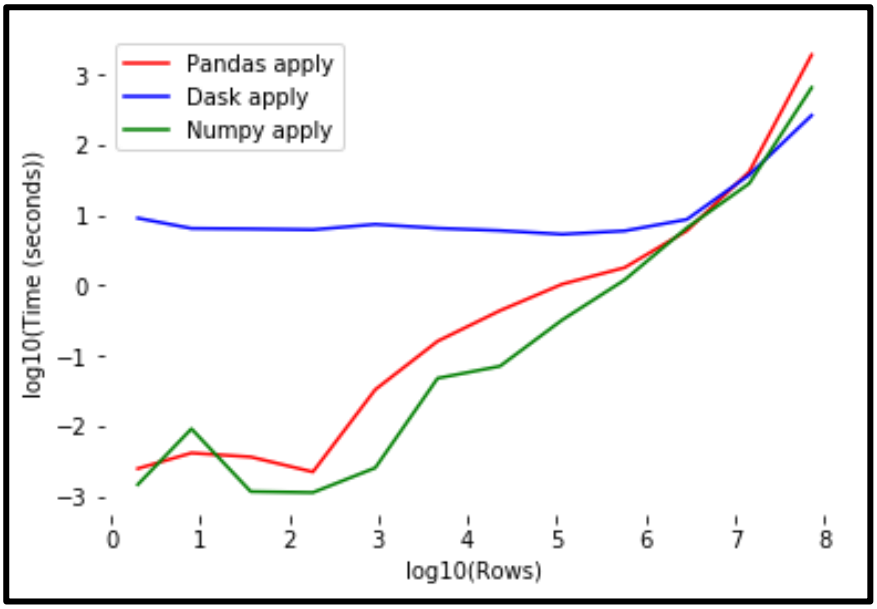

Figure 37 : Test 3 temps d'exécution des librairies sur serveur 4cores

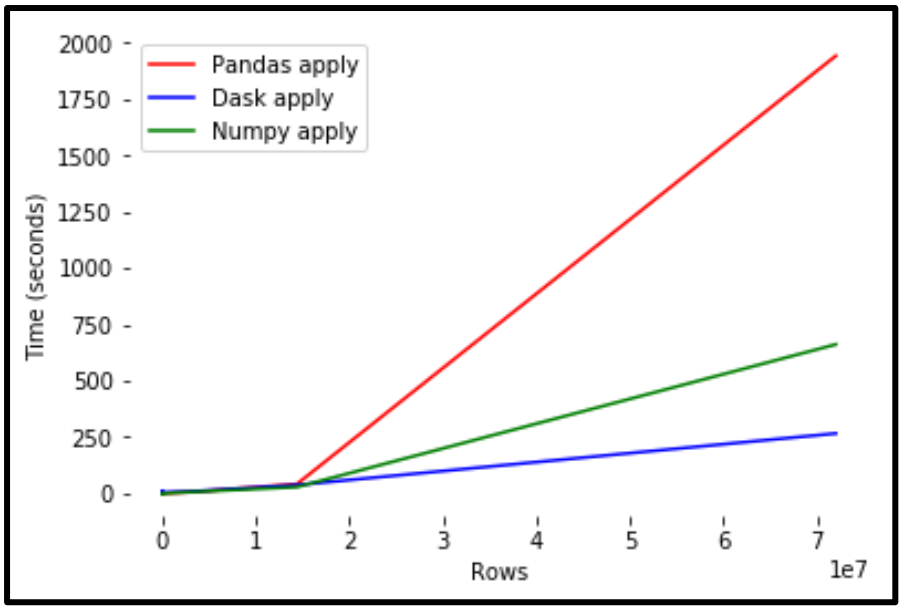

Figure 38 : Test 3 temps d'exécution des librairies sur MacBook Pro

- **Dell R440 8cores :**

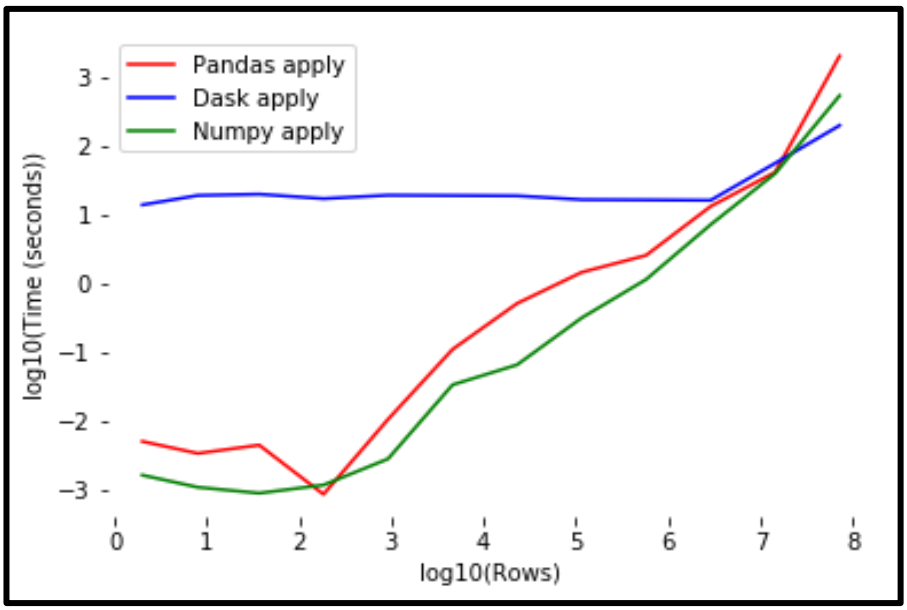

Figure 39 : Test 3 temps d'exécution des librairies sur server

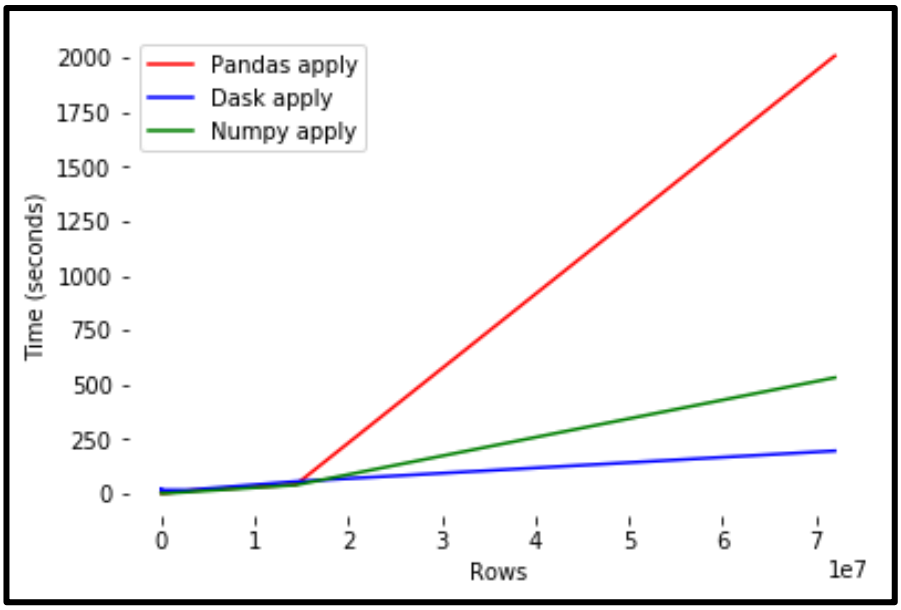

Figure 40 : Test 3 temps d'exécution des librairies sur MacBook Pro

On remarque que le Macbook Pro est plus performant que le serveur Dell PowerEdge R440, cela dû à plusieurs facteurs :

- **Le type de disque dur**, le Macbook Pro est équipé d'un disque dur SSD qui est plus rapide et performant en lecture et écriture que le HDD du serveur Dell.
- **La virtualisation**, on ne peut pas utiliser toute la puissance du CPU et de la Ram car une partie est réservée automatiquement pour le système VmWare qui gère le serveur

On remarque aussi que la bibliothèque Dask est plus performante que Pandas et NumPy.

Dask permet de traiter des données plus grandes que la RAM en utilisant la notion du parallélisme, il permet de manipuler les données d'une façon partitionnée entre la Ram et le disque dur, et utilise tous les cores du CPU.

III.2.3.2 Applications 2

Dans la première application, nous avons constaté que la bibliothèque Dask est plus performante que NumPy et Pandas.

Pour vérifier ce résultat, on a utilisé une base de données générer aléatoirement par les trois bibliothèques et on a calculé le temps nécessaire pour sa production.

On a obtenu les résultats suivants :

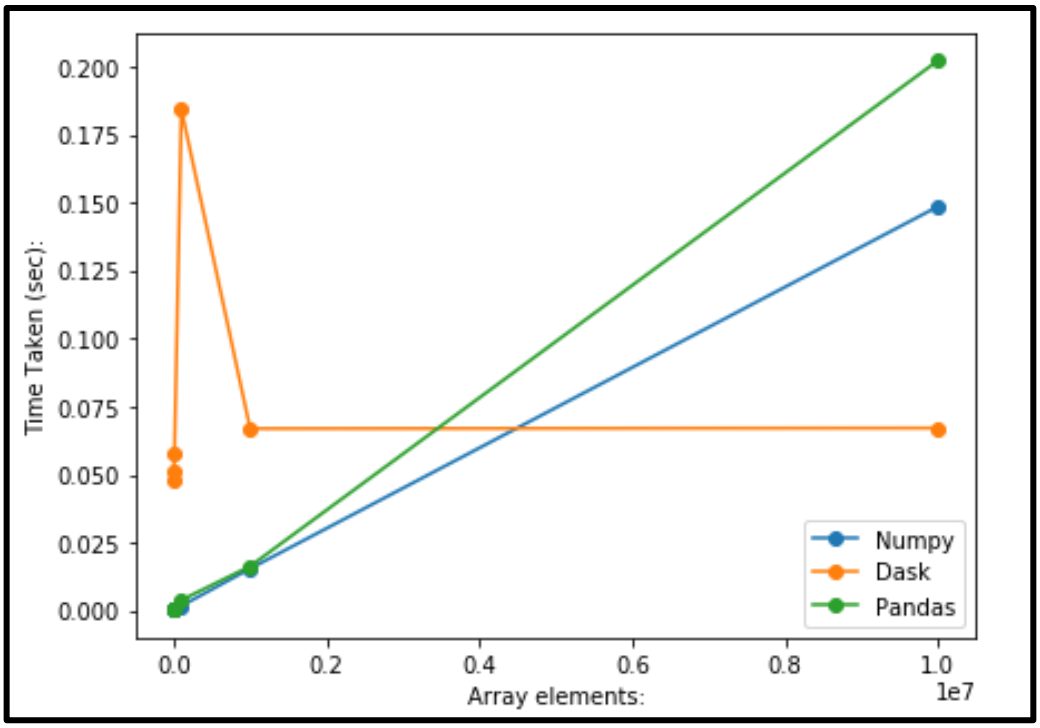

Figure 41 : Le temps de génération des éléments de taille 1e7

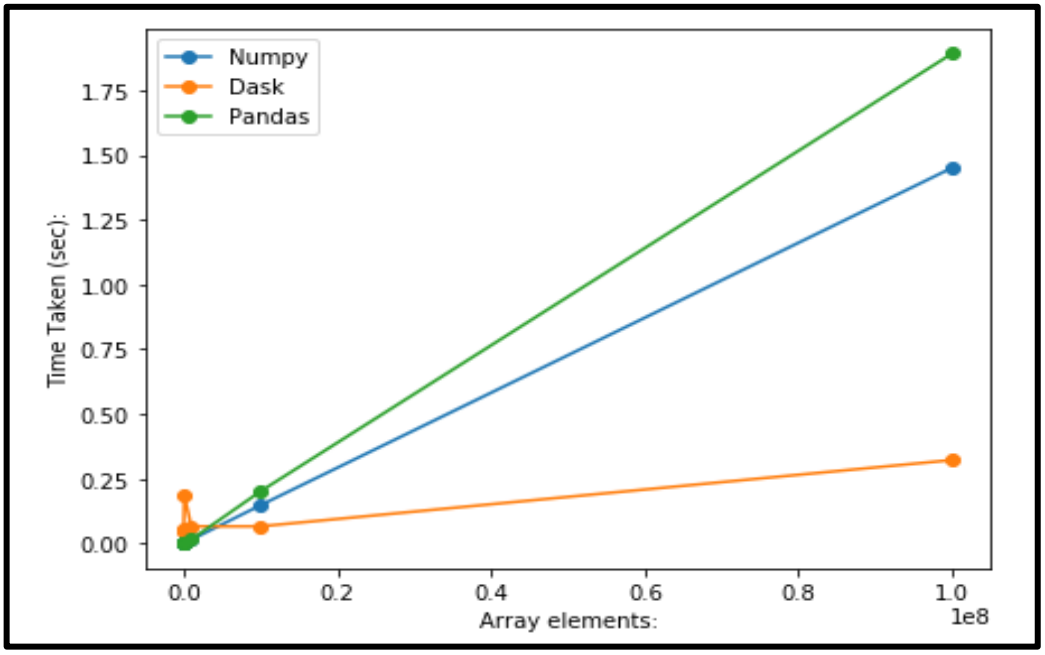

Figure 42 : Le temps de génération des éléments de taille 1e8

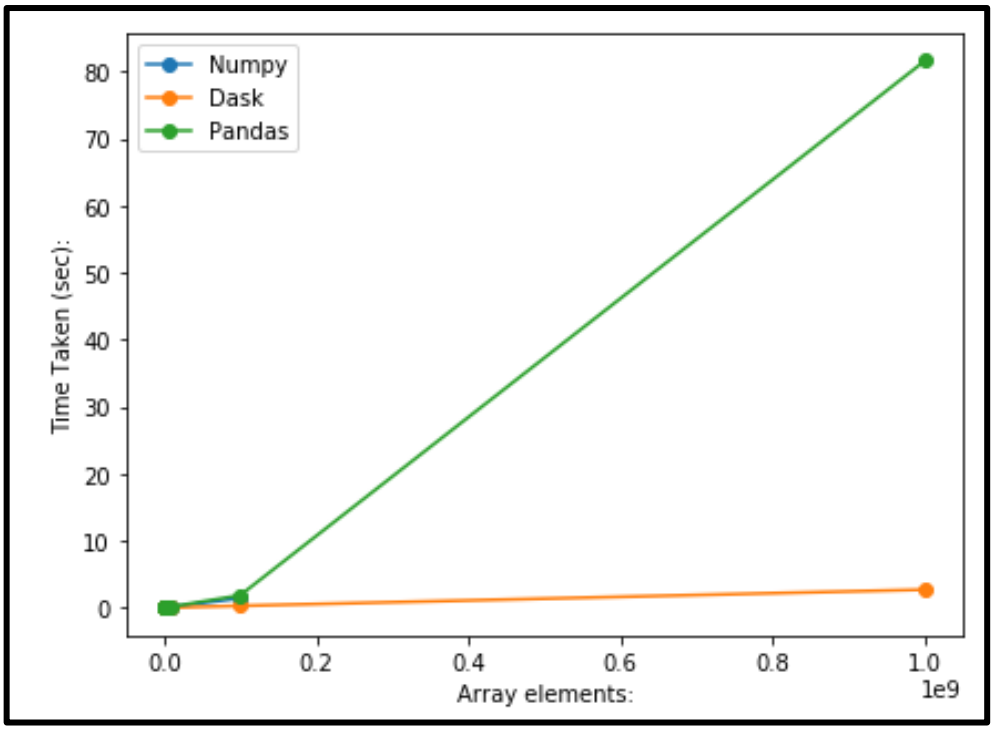

Figure 43 : Le temps de génération des éléments de taille 1e9

NumPy est plus rapide que Dask et Pandas pour un plus petit nombre d'éléments, Dask reprenant NumPy et Pandas pour environ 1e7 éléments. NumPy incapable de produire des résultats pour un plus grand nombre d'éléments car il n'est pas en mesure de les mettre sur la mémoire.

Alors Pandas est plus performant que NumPy, et Dask est la bibliothèque la plus performante.

III.3 Application IoT de mesure de température

L'IoT envahit petit à petit notre quotidien. Des capteurs nous renseignent sur notre environnement, de nouvelles possibilités d'interactions avec les objets apparaissent. Puisqu'on a présenté dans ce rapport l'état de l'art de l'IoT, on a décidé de créer une application IoT de mesure de température.

Cette application donne la température pour chaque seconde selon la durée saisie par l'utilisateur.

Nous allons voir les étapes de création de cette application :

III.3.1 Le Matériel

Phidget SBC3 :

Le PhidgetSBC3 est un ordinateur à une seule carte avec une interface 8/8/8 intégré. À sa base, On peut le connecter à l'aide d'un câble réseau au lieu d'USB. Le PhidgetSBC3 fournit également six ports à haute vitesse qui nous permettent d'utiliser des ports USB normaux.

Le PhidgetSBC3 expose une interface facile à utiliser pour configurer et exécuter des applications personnalisées à bord.

Le PhidgetSBC est un ordinateur embarqué qui exécute Debian GNU/Linux. Il fournit un accès complet au shell via un serveur SSH intégré, un accès au dépôt de paquets Debian complet et à tous les outils de ligne de commande standard attendus sur un système Linux moderne.

Un PhidgetInterfaceKit 8/8/8 intégré permet de connecter des appareils à l'une des 8 entrées analogiques, 8 entrées numériques et 8 sorties numériques.

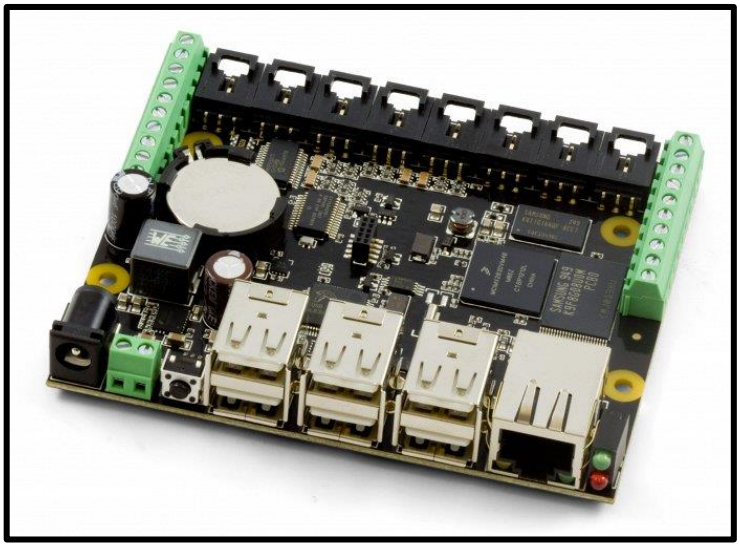

Figure 44 : Phidget SBC3

Le capteur de température :

On a utilisé le capteur E209436 qui peut fonctionner sous température entre -1C° et 125C° et jusqu'à 600V.

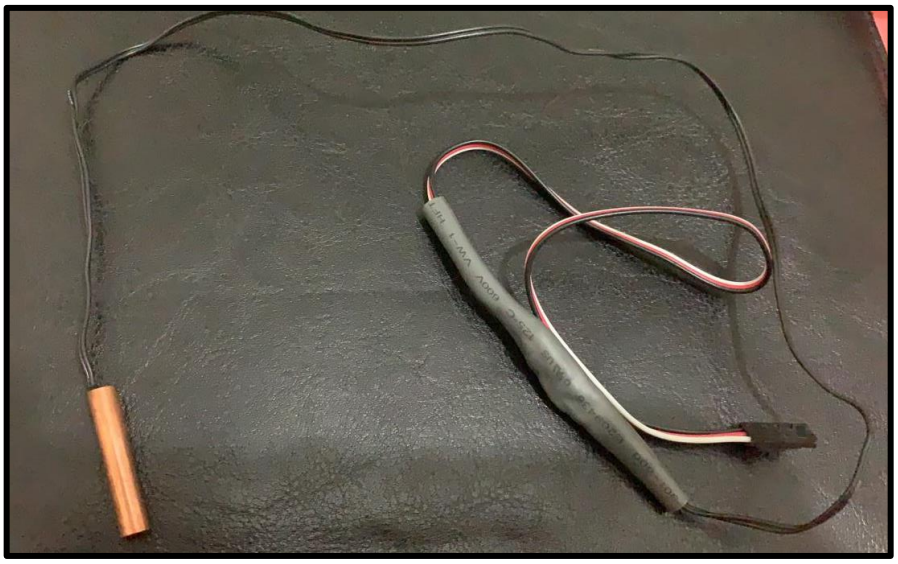

Figure 45 : Capteur E209436 de température

Clé wifi :

Pour connecter PhidgetSBC3 à notre réseau local

Figure 46 : Clé WiFi

III.3.2 Le montage

On a connecté le capteur au PhidgetSBC3 à partir de l'entrée analogique et la clé WiFi par un port USB.

Pour la première utilisation, on doit connecter le Phidget par un câble réseau à notre routeur afin de configurer l'accès WiFi, on peut également garder la connexion Ethernet.

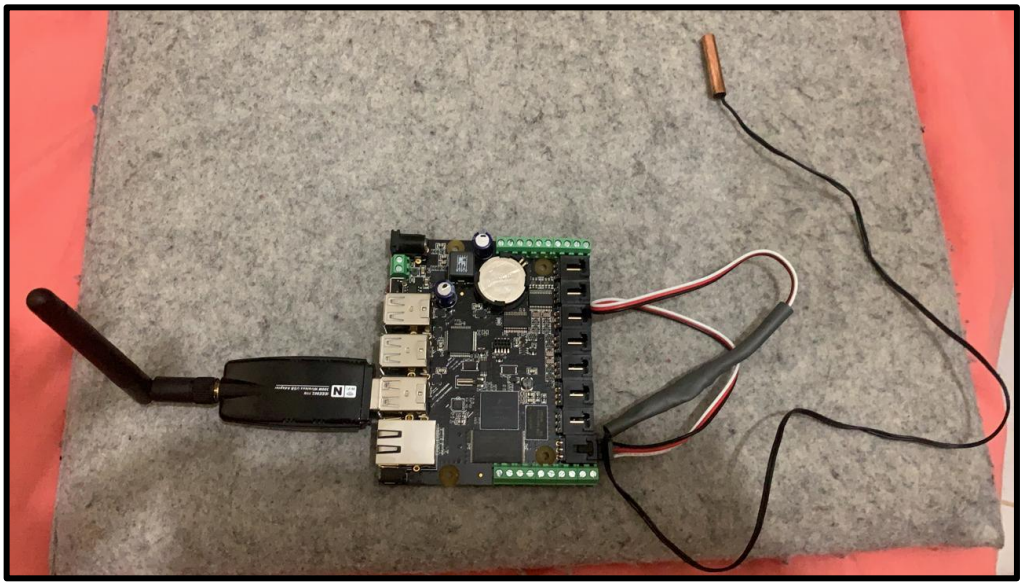

Figure 47 : Montage de Phidget et capteur

III.3.3 L'environnement du développement

Pour le développement notre application, on a utilisé le langage Python, qui est supporté par le PhidgetSBC3. Notre capteur n'est pas reconnu par Phidget comme un capteur de température, il est considéré comme un capteur de tension, donc on a utilisé les API VoltageRatioInput fournit par le fabricant, et on a étalonné les valeurs reçus pour les convertir en C°. Les résultats sont approximatifs.

III.3.4 L'application

- Pour commencer les mesures, on saisit d'abord la durée des mesures en secondes et on clique sur commencer les mesures :

Figure 48 : Application étape 1

- On attend que la durée se termine et on clique sur afficher les résultats pour voir les valeurs reçus :

Figure 49 : Application étape 2

- On peut sauvegarder les résultats sous format csv en cliquant sur sauvegarder, un fichier .csv se génère automatiquement dans le dossier contenant l'application.

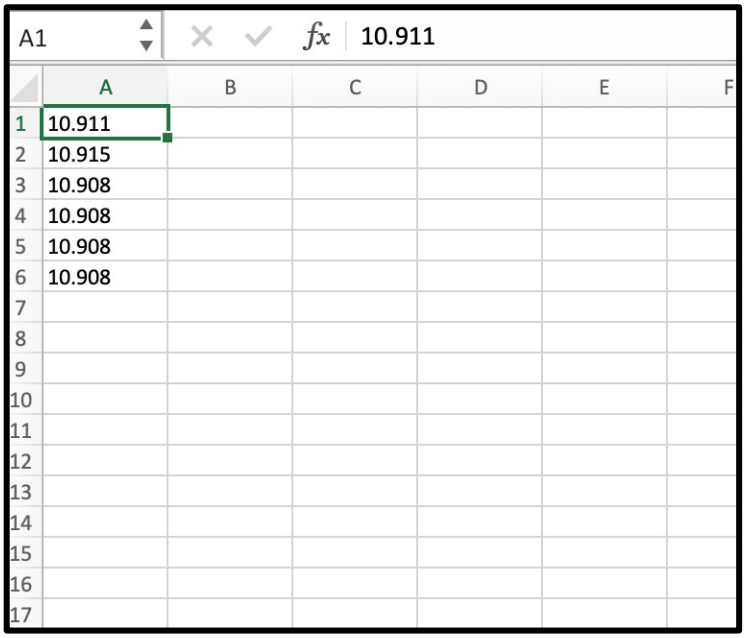

Figure 50 : Application résultats

Nous avons réussi à configurer le nouveau serveur du laboratoire LSSC, après avoir effectué plusieurs tests, on a remarqué qu'il est moins puissant que le MacBook Pro. Ces tests nous ont permis de déterminer la librairie python la mieux adapter pour traiter les données.
Nous avons aussi réussi à développer une application IoT de mesures de température.

Conclusion générale

Elles existent plusieurs définitions de l'IoT, des définitions basées sur l'architecture, et d'autres sur les fonctionnalités, mais jusqu'à présent elle n'existe aucune définition standard. Il en est de même pour l'architecture, chaque application IoT peut avoir sa propre architecture pour optimiser ses performances, cependant l'architecture de trois couches : Matériels-Middleware et Présentation peut être considéré comme référence.

Les domaines d'application de l'IoT sont nombreux, et ses défis le sont aussi, surtout la matière de sécurité qui représente le défi majeur. Les objets produisent de grandes quantités d'informations, et sont connectés via un réseau, or les réseaux sont vulnérables aux attaques. Donc il faut assurer la protection des données. Les défis de l'IoT motivent les chercheurs. On trouve plusieurs articles qui traitent soit l'architecture soit un domaine d'application.

Nous avons configuré aussi le nouveau serveur du laboratoire LSSC, qui va être utilisé pour des applications IoT et les calculs scientifiques ou simulations… Après différents tests, nous avons remarqué que la virtualisation diminue ses performances, il est conseillé d'installer un système directement.

Nous terminons cette conclusion en évoquant les différentes perspectives de recherche que nous envisageons d'aborder dans le futur :

- Améliorer la sécurité de l'IoT, dans une couche ou plusieurs.
- Concevoir un algorithme de traitement Big Data performant quel que soit le volume des données.
- Améliorer l'application IoT développé dans le cadre de notre travail en ajoutant d'autres fonctionnalité

Bibliographie

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, *Vision and challenges for realising the Internet of Things: CERP-IoT – Cluster of European research projects on the Internet of Things*, no. March. 2010.

[3] A. Juels, "RFID security and privacy: A research survey," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 381–394, 2006.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Akyildiz et al._2002_Wireless sensor networks a survey.pdf," vol. 38, pp. 393–422, 2002.

[5] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive Mob. Comput.*, vol. 4, no. 3, pp. 303–334, 2008.

[6] H. Alzaid, E. Foo, and J. M. Gonzalez Nieto, "Secure Data Aggregation in Wireless Sensor Network: a survey BT - Sixth Australasian Information Security Conference (AISC 2008)," vol. 81, no. January, pp. 93–105, 2008.

[7] S. Mora, F. Gianni, and M. Divitini, "RapIoT toolkit: Rapid prototyping of collaborative Internet of Things applications," *Proc. - 2016 Int. Conf. Collab. Technol. Syst. CTS 2016*, pp. 438–445, 2016.

[8] V. Gazis *et al.*, "Short Paper : IoT : Challenges , Projects , Architectures," pp. 145–147, 2015.

[9] H. D. Ma, "Internet of things: Objectives and scientific challenges," *J. Comput. Sci. Technol.*, vol. 26, no. 6, pp. 919–924, 2011.

[10] B. Billet and S. De, "Système de gestion de flux pour l ' Internet des objets intelligents To cite this version : ' I Data Stream Management System for the Future," 2015.

[11] H. Zhang and L. Zhu, "Internet of Things: Key technology, architecture and challenging problems," *Proc. - 2011 IEEE Int. Conf. Comput. Sci. Autom. Eng. CSAE 2011*, vol. 4, pp. 507–512, 2011.

[12] S. Helal, S. De Deugd, R. Carroll, K. E. Kelly, B. Millett, and J. Ricker, "Standards & Emerging Technologies SODA: Service-Oriented Device Architecture MODELING DEVICES AS SERVICES," 2006.

[13] M. Buettner, B. Greenstein, A. Sample, J. R. Smith, and D. Wetherall, "Revisiting Smart Dust with RFID Sensor Networks," *Proc. 7th ACM Work. Hot Top. Networks*, 2008.

[14] C. Floerkemeier, C. Roduner, and M. Lampe, "RFID Application Development With the Accada Middleware Platform," *IEEE Syst. J.*, vol. 1, no. 2, pp. 82–94, 2007.

[15] E. Welbourne *et al.*, "Building the Internet of Things Using RFID," 2009.

[16] C. Perera, P. P. Jayaraman, A. Zaslavsky, D. Georgakopoulos, and P. Christen, "Sensor discovery and configuration framework for the Internet of Things paradigm," *2014 IEEE World Forum Internet Things, WF-IoT 2014*, pp. 94–99, 2014.

[17] S. Clayman and A. Galis, "Inox," pp. 1–8, 2011.

[18] G. K. Eleftherakis, "Bilattices and Morita Equivalence of MASA Bimodules," *Proc. Edinburgh Math. Soc.*, vol. 59, no. 3, pp. 605–621, 2016.

[19] J. C. Zhao, J. F. Zhang, Y. Feng, and J. X. Guo, "The study and application of the IOT technology in agriculture," *Proc. - 2010 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol.*

ICCSIT 2010, vol. 2, pp. 462–465, 2010.

[20] P. P. Ray, "Towards an internet of things based architectural framework for defence," *2015 Int. Conf. Control Instrum. Commun. Comput. Technol. ICCICCT 2015*, pp. 411–416, 2016.

[21] G. Yang *et al.*, "A Health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2180–2191, 2014.

[22] P. P. Ray, "IoT): An Architectural Framework for Monitoring Health of Elderly People," *Int. Conf. Sci. Eng. Manag. Res.*, pp. 3–5, 2014.

[23] L. Yu, Y. Lu, and X. J. Zhu, "Smart hospital based on internet of things," *J. Networks*, vol. 7, no. 10, pp. 1654–1661, 2012.

[24] Y. J. Fan, Y. H. Yin, L. Da Xu, Y. Zeng, and F. Wu, "IoT-based smart rehabilitation system," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1568–1577, 2014.

[25] Ghose, A., Biswas, P, Bhaumik, C., Sharma, M., "No Title," in *Road condition monitoring and alert application: using in-vehicle Smartphone as Internet-connected sensor. In: Proceedings of Pervasive Computing and Communications Workshops (PERCOM Workshops),* 2012, pp. 489–491.

[26] M. Wang, P. Guo, and Y. Jia, "Comprehensive evaluation on the regional risk of group events," *Asia-Pacific Power Energy Eng. Conf. APPEEC*, pp. 0–3, 2011.

[27] C. Networks *et al.*, "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Networks*, vol. 61, no. November, pp. 217–238, 2014.

[28] A. Zhang, J., "The application of internet of things (IOT) in emergency management system in China. Security (HST)," in *The application of internet of things (IOT) in emergency management system in China. In: Proceedings of IEEE International Conference on Technologies for Homeland Security (HST),* 2010, pp. 139–142.

[29] S. Mosser, F. Fleurey, B. Morin, F. Chauvel, A. Solberg, and I. Goutier, "SENSAPP as a reference platform to support cloud experiments: From the internet of things to the internet of services," *Proc. - 14th Int. Symp. Symb. Numer. Algorithms Sci. Comput. SYNASC 2012*, pp. 400–406, 2012.

[30] P. P. Ray, "Generic Internet of Things architecture for smart sports," *2015 Int. Conf. Control Instrum. Commun. Comput. Technol. ICCICCT 2015*, pp. 405–410, 2016.

[31] W. Yang, B., Nie, X., Shi, H., Gan, "M-learning mode research based on internet of things.," in *M-learning mode research based on internet of things. In: Proceedings of International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, 2011, pp. 5623–5627.

[32] Z. Jiang, L., Xu, L.D., Cai, H., Jiang, "An IoT-oriented data storage framework in cloud computing platform.," in *An IoT-oriented data storage framework in cloud computing platform. IEEE Trans. Industr.*, 2014, pp. 1443–1451.

[33] N. Narendra, K. Ponnalagu, A. Ghose, and S. Tamilselvam, "Goal-Driven Context-Aware Data Filtering in IoT-Based Systems," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2015-Octob, pp. 2172–2179, 2015.

[34] N. Shahid and S. Aneja, "Internet of Things: Vision, application areas and research challenges," *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, vol. 10, no. 7, pp. 583–587, 2017.

[35] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[36] M. U.Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A Critical Analysis on the Security

Concerns of Internet of Things (IoT)," *Int. J. Comput. Appl.*, vol. 111, no. 7, pp. 1–6, 2015.
[37] K. El Emam, "Protecting privacy using k-anonymity (Appendix A : Risk Estimates)," *J. Am. Med. Informatics*, vol. 15, no. 5, pp. 1–5, 2008.
[38] A. Wahab, O. Ahmad, M. Muhammad, and M. Ali, "A Comprehensive Analysis on the Security Threats and their Countermeasures of IoT," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, 2017.
[39] T. Karygiannis, B. Eydt, G. Barber, L. Bunn, and T. Phillips, "Guidelines for securing radio frequency identification (RFID) systems-Recommendations of the," *Natl. Inst. Stand. …*, 2007.
[40] Z. Xu, Y. Yin, and J. Wang, "A Density-based Energy-efficient Clustering Algorithm for Wireless Sensor Networks," *Int. J. Futur. Gener. Commun. Netw.*, vol. 6, no. 1, pp. 75–86, 2013.
[41] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.
[42] R. Uttarkar and P. R. Kulkarni, "Internet of Things : Architecture and Security," *Int. J. Comput. Appl.*, vol. 3, no. 4, pp. 12–19, 2014.
[43] M. Burmester and B. De Medeiros, "RFID Security: Attacks, Countermeasures and Challenges," *Comput. Sci. Dep. Florida State Univ. Retrieved Novemb.*, vol. 21, p. 10, 2007.
[44] B. Khoo, "RFID As an enabler of the internet of things: Issues of security and privacy," *Proc. - 2011 IEEE Int. Conf. Internet Things Cyber, Phys. Soc. Comput. iThings/CPSCom 2011*, pp. 709–712, 2011.
[45] A. Mitrokotsa, M. R. Rieback, and A. S. Tanenbaum, "Classification of RFID Attacks," pp. 73–86, 2011.
[46] X. M. Li, L. Q. Ge, Z. J. Xin, and C. Chen, "Study on Security Problems of the Internet of Things," *Appl. Mech. Mater.*, vol. 303–306, no. Mic, pp. 2425–2428, 2013.
[47] K. Zhao and L. Ge, "A survey on the internet of things security," *Proc. - 9th Int. Conf. Comput. Intell. Secur. CIS 2013*, pp. 663–667, 2013.
[48] T. Bhattasali, R. Chaki, and S. Sanyal, "Sleep Deprivation Attack Detection in Wireless Sensor Network," *Int. J. Comput. Appl.*, vol. 40, no. 15, pp. 19–25, 2012.
[49] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based internet of things," *Int. J. Distrib. Sens. Networks*, vol. 2013, 2013.
[50] A. Daniel, "A Survey on Detection of Sinkhole Attack in Wireless Sensor Networks," vol. 91, no. 7, pp. 48–52, 2014.
[51] Z. K., L. X., L. R., and S. X., "Sybil attacks and their defenses in the internet of things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, 2014.

Internet of things : état de l'art et applications

Résumé

L'Internet of things (IoT) rend les objets qui nous entourent intelligents en les connectant via un réseau. L'IoT est un domaine d'actualité qui forme un écosystème d'objets, de communications, d'applications et d'analyses des données.

Dans ce travail nous avons essayé d'une part de définir l'IoT, présenter son architecture, ses domaines d'applications, ses défis et les différents travaux de recherche. La sécurité représente un problème majeur de l'IoT, nous avons donné ses problématiques, les techniques utilisées et aussi ses défis.

D'autre part nous avons configuré un serveur Dell PowerEdge R440 pour utilisation futur de développement d'application IoT, puis nous avons comparé ses performances avec d'autres ordinateurs.

Nous avons aussi développé une application de mesure de température à l'aide d'ordinateur monocarte PhidgetSBC3 et un capteur de température. Cette application nous donne la température pour chaque seconde dans une durée déterminée par l'utilisateur.

Mots clés : internet of things, Dell PowerEdge

Internet of things: state of the art and applications

Abstract

The Internet of things (IoT) makes objects around us smart by connecting them via a network. IoT is a topical field that forms an ecosystem of objects, communications, applications and data analysis.

In this work we tried on the one hand to define IoT, to present its architecture, its fields of applications, its challenges and the different works of researchers. Security is a major problem of IoT, we gave its problems, the techniques used and also its challenges.

On the other hand, we configured a Dell PowerEdge R440 server for future use in IoT application development, and then compared its performance with other computers.

We have also developed a temperature measurement application using a PhidgetSBC3 monocarte computer and a temperature sensor. This application gives us the temperature for each second within a duration determined by the user.

Keywords: internet of things, Dell PowerEdge

Master Systèmes Intelligents & Réseaux
Département d'Informatique
Faculté des Sciences et Techniques de Fès
A.U. 2018 - 2019