Numéro d'ordre : 28/IRS/TCO Année Universitaire 2013/2014

UNIVERSITE D'ANTANANARIVO

ECOLE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT TELECOMMUNICATION

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention

Du diplôme de Master

Titre : Ingénieur

Mention: Télécommunication

Parcours: Ingénierie des Réseaux et Systèmes

Par : **RAMBELOSON Andriantomefisoa**

EVALUATION DE PERFORMANCES DE CALCUL ET DE STOCKAGE D'UN CLOUD PRIVE

Soutenu le 26 Mars 2015 devant la commission d'examen composée de :

Président :

M. ANDRIAMIASY Zidora

Examinateurs :

M. RANDRIAMITANTSOA Paul Auguste

Mme RAMAFIARISONA Hajasoa Malalatiana

Mme ANDRIANTSILAVO Haja

Directeur de mémoire :

M. BOTO ANDRIANANDRASANA Jean Espérant

REMERCIEMENTS

Au terme de la rédaction de ce présent ouvrage, je remercie en premier lieu Jésus Christ qui m'a donné le courage et la joie durant les cinq dernières années passées au sein de l'ESPA.

Je remercie Monsieur ANDRIANARY Philippe, Professeur Titulaire et Directeur de l'Ecole Supérieure Polytechnique d'Antananarivo, de m'avoir accueilli au sein de son établissement.

Mes sincères remerciements s'adressent également à Monsieur RAKOTOMALALA Mamy Alain, Maître de Conférences et Chef de Département Télécommunication, qui a consacré ses efforts pour le bon déroulement de notre formation au sein du département.

J'exprime mes profondes gratitudes à Monsieur BOTO ANDRIANANDRASANA Jean Espérant, Assistant d'Enseignement et de Recherche, pour m'avoir dirigé tout au long de ce travail, et qui n'a cessé de me prodiguer de précieux conseils.

Je tiens à remercier aussi Monsieur ANDRIAMIASY Zidora, Maître de conférences, qui me fait l'honneur de présider les membres du Jury de ce mémoire.

Je remercie chaleureusement tous les membres du jury qui ont bien voulu consacrer une partie de leur temps, pour examiner et améliorer l'ensemble de ce travail :

- M. RANDRIAMITANTSOA Paul Auguste, Professeur Titulaire ;
- Mme. RAMAFIARISONA Hajasoa Malalatiana, Maître de Conférences ;
- Mme. ANDRIANTSILAVO Haja, Assistant d'Enseignement et de Recherche.

J'exprime mes vifs remerciements à toute ma famille pour leur soutient morale, financière et spirituel. Sans votre aide ce travail est inachevable. Je suis également reconnaissant envers mes amis, mes collègues, mes frères et sœurs, ainsi que tous ceux qui ont contribué, de près ou de loin, à l'élaboration de ce travail. Je vous remercie tous et que Dieu tout puissant vous bénisse.

TABLE DES MATIERES

NOTATIONS ET ABREVIATIONS

Notations mathématiques

$-V_{high}$	Tension d'écriture d'un FGMOS
$C_{mém}$	Capacité mémoire
$Débit_{cache}$	Débit vers la mémoire cache
$Débit_{E/S}$	Débit de transfert de données en entrée/sortie
D_{sys}	Dimension d'un système d'équations
E_{IOPS}	Nombre d'opération d'entrée par seconde.
$\overline{E_{IOPS}}$	Nombre d'opération d'entrée moyenne par seconde
$f_{horloge}$	Fréquence d'horloge
$L_{rotation}$	Latence de rotation
M_{disp}	Mémoire vive disponible
$N_{\text{cœurs}}$	Nombre de cœurs
N_{enc}	Nombre de bits utiles pour un encodage
$Nombre_{IOPS}$	Nombre d'opération entrée/sortie par second
N_{op}	Nombre d'opération
$N_{opération/cycle}$	Nombre d'opération par cycle
$P_{théorique}$	Puissance théorique
$R_{QEMU,\ U\ Serveur}$	Rendement de virtualisation de QEMU avec Ubuntu Server
$R_{QEMU,\ WIn\ XP}$	Rendement de virtualisation de QEMU avec Windows Xp
S_{IOPS}	Nombre d'opération de sortie par seconde
$\overline{S_{IOPS}}$	Nombre d'opération de sortie moyenne par seconde
$T_{accès}$	Temps d'accès
$Taille_{bloc}$	Taille d'un bloc de données
$T_{E/S}$	Temps d'entrée/sortie
$T_{recherche}$	Temps de recherche
V_{EA}	Vitesse d'écriture aléatoire
V_{ES}	Vitesse d'écriture séquentielle
$\overline{V_{ES}}$	Vitesse d'écriture séquentielle
V_{high}	Tension de réinitialisation d'un FGMOS

V_{in}	Tension de lecture d'un FGMOS
V_{LA}	Vitesse de lecture aléatoire
V_{LS}	Vitesse de lecture séquentielle
$\overline{V_{LS}}$	Vitesse de lecture séquentielle moyenne
V_{off}	Tension de blocage d'un FGMOS
V_{on}	Tension de conduction d'un FGMOS

Abréviations techniques

3D	Trois Dimension
AAM	Acoustic Mismatch Model
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
BOINC	Berkeley Open Infrastructure for Network Computing
C/S	Client/Serveur
CDMI	Cloud Data Management Interface
CPU	Central Processing Unit
CDROM	Compact Disk - Read Only Memory
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
D	Drain
DNS	Domain Name System
E/S	Entrée/Sortie
EC2	Elastic Compute Cloud
EGEE	Enabling Grid for E-sciencE
FGMOS	Floating-Gate MOSFET
FLOPS	Floating Point Operations Per Second
G	Gate
GFLOPS	Giga FLOPS
GHz	Giga Hertz
Go	Gigaoctet
GPS	Global Positionning System
GUI	Graphical User Interface
HDD	Hard disk drive
HP	Hewlett-Packard
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IBM	International Business Machines

IEEE	Institute of Electrical and Electronics Engineers
IOPS	Input/Output Operations Per Second
IP	Internet Protocol
ISO	International Standards Organisation
IT	Information Technology
JMS	Java Message Service
JPMorgan	John Pierpont Morgan
Ko	Kilooctet
MLC NAND	Multi Level Cell Negative-AND
Mo	Megaoctet
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
ms	milliseconde
NAND	Negative-AND
NASA	National Aeronautics and Space Administration
NIST	North American National Institute for Standard and Technology
NTP	Network Time Protocol
OASIS	Organization for the Advancement of Structured Information Standards
OS	Operating System
OVFM	Open Virtual Machine Format
P2P	Peer-to-Peer
PaaS	Platform as a Service
PC	Personal Computer
PDF	Portable Document Format
PFLOPS	Peta FLOPS
PHP	Personal Home Page
PUE	Power Usage Effectiveness
QCOW2	Qemu Copy-On-Write version 2
QEMU	Quick Emulator
RabbitMQ	Rabbit Message Queuing
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory

RBAC	Role Based Acces Control
RFC	Requests For Comments
RPC	Remote Procedure Call
RPM	Rotation Par Minute
RPO	Recovery Point Objective
RTO	Recovery Time Objective
S	Source
SaaS	Software as a Service
SETI	Search for ExtraTerrestial Inteligence
SGBD	Système de Gestion de Base de Données
SI	Système d'Information
SLC NAND	Single Level Cell Negative-AND
SMART	Self-Monitoring, Analysis, and Reporting Technology
SNIA	Storage Networking Industry Association
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSD	Solid-State Drive
t	temps
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/ Internet Protocol
UAL	Unité arithmétique et logique
UDDI	Universal Description Discovery and Integration
UTC	Universal Time Coordinated
VM	Virtual Machine
VNC	Virtual Network Computing
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WebOS	Web Operating System
WSDL	Web Services Description Language
XML	eXtensible Markup Language

INTRODUCTION GENERALE

L'actuelle émergence de l'informatique à la demande est une preuve de l'évolution très significative des technologies de communication, qui sont devenues facilement accessibles tant aux simples usagers qu'aux grandes entreprises. Cela rend les ressources informatiques indispensables dans la vie quotidienne. L'achat et l'ajout des nouvelles unités de calcul et de stockage se font à chaque fois où on atteint la capacité maximale. Le Cloud Computing révolutionne cette méthode classique de l'environnement informatique. C'est le moyen pour chaque individu ou entreprise de disposer, partout et à moindre coût des services informatiques tout en permettant une adaptation immédiate des ressources selon les besoins. Néanmoins, la limite se repose seulement sur la capacité du client à payer. De plus, les services Cloud sont disponibles à la demande et facturés selon le principe du paiement à l'usage, idéals pour le contrôle des dépenses. Ainsi, il n'implique aucun investissement en matériel supplémentaire pour les clients.

De ce fait, plusieurs entreprises courent vers le marché Cloud, tout en décrivant leurs services comme plus performants et idéals pour les clients. Toutefois, le besoin de croissance des utilisateurs en termes de quantité et de qualité peut représenter un problème pour les administrateurs qui ne se soucient que de donner une satisfaction à ses clients. Un Cloud privé est avantageux pour le commencement d'une étude de performance, on peut le déployer sur quelques serveurs physiques. La maitrise d'un tel type de Cloud ouvre une voie vers tous les autres types.

Pour faire face à un tel défi, ce qui vient naturellement à l'esprit est l'amélioration, voire le surdimensionnement des infrastructures existantes, d'une manière explicite, empiler plusieurs serveurs physiques pour répondre aux demandes de ressources. Des outils fiables sont aussi recommandés pour la mesure de la capacité totale du système établi, ainsi que pour savoir le rendement de performance pour chaque ajout de serveur. Le but final est d'assurer que tous les clients reçoivent leurs quotas. Cela nous amène au titre : « Evaluation de performances de calcul et de stockage d'un Cloud privé ».

Ce présent ouvrage détaillera en premier lieu chaque architecture informatique et technologie à l'origine du Cloud Computing. On présentera en second lieu le concept du Cloud Computing qui englobe les modes de déploiement et les différents types de services. Ensuite, une étude de performance de calcul et de stockage d'un système informatique standard sera faite. On commencera par la définition des unités servant à la classification de chaque équipement

informatique selon leur capacité en ressource, suivi d'une étude d'un système doté d'un processeur et un disque de stockage jusqu'à la théorie de la performance d'un système multiprocesseur et multidisque de stockage. C'est à la fin de cette troisième partie qu'on abordera les outils de mesure de performance de calcul et de performance de stockage. La dernière partie sera consacrée à la présentation de l'outil de base pour un déploiement d'un Cloud privé, suivi d'une réalisation sur deux serveurs. Une analyse de la performance du Cloud déployé sera menée, les résultats seront présentés d'une part et interprétés d'autre part. On essaiera d'expliquer l'impact de cette dernière sur la performance du système.

CHAPITRE 1

LES ARCHITECTURES ET TECHNOLOGIES A L'ORIGINE DU CLOUD COMPUTING

1.1 Introduction

Le Cloud Computing est la cinquième génération de l'architecture informatique, et comme chaque génération, celle-ci ne déroge pas la règle puisqu'elle veut révolutionner l'IT (Information Technology) d'entreprise dans l'intégralité du modèle. C'est une vaste ambition mais impossible sans l'existence des quatre autres générations antérieures.

En effet, la première génération voyait le jour en 1970, baptisait « MainFrame ». Peu après, l'ère du système distribué a commencé. L'architecture Client/Serveur apparait en 1980. L'augmentation du nombre d'utilisateurs et de la quantité d'informations sur internet accélérait la standardisation de l'architecture Web en 1990. Ce dernier basculait vers le domaine commercial et entrainait la naissance de l'architecture orientée service ou SOA (Service Oriented Architecture) en 2000.

A part, les architectures informatiques, la technologie de virtualisation et le système Grid sont aussi des piliers du Cloud Computing. Tout au long de ce chapitre, nous allons détailler ces architectures et technologies à l'origine du Cloud Computing.

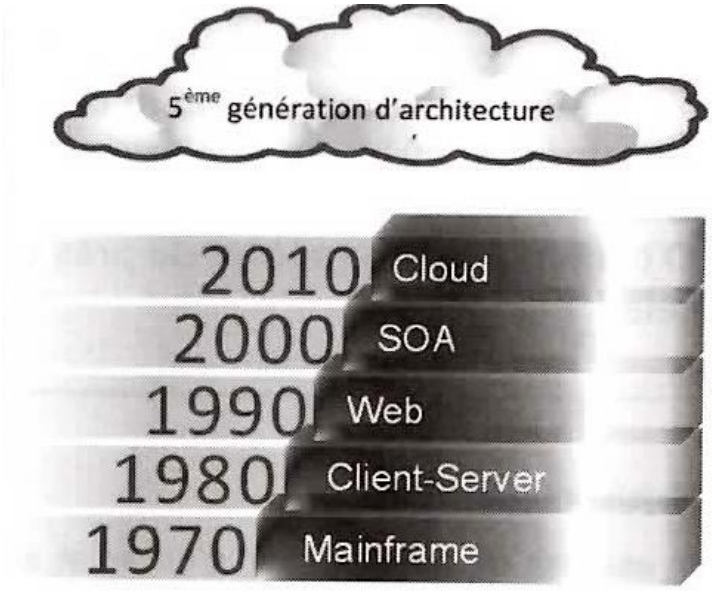

Figure 1.01: *Générations d'architecture informatique [1]*

1.2 Architecture Client/Serveur (C/S)

On parle de C/S quand il s'agit d'un système constitué de plusieurs ordinateurs distants reliés par un réseau, et voulant s'inter-échanger des données, selon une certaine hiérarchie, en parlant un même langage communément appelé protocole. Dans cette soi-disant « hiérarchie », celui qui effectue la requête est appelé le client, tandis que celui qui retourne une réponse à cette requête est appelé le serveur [2].

1.2.1 Modèle général

Le modèle général d'un système à architecture client-serveur est présenté par la figure ci-dessous.

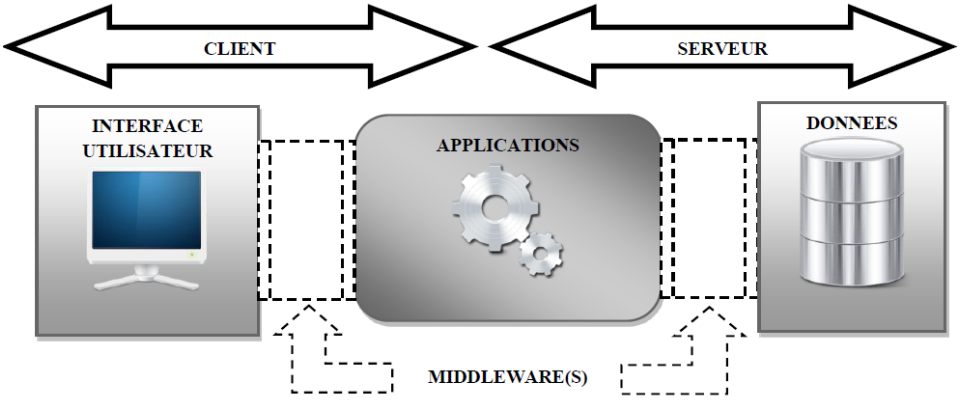

Figure 1.02: *Modèle général de l'architecture système*

1.2.1.1 Interface utilisateur

Les clients accèdent à l'ensemble du système grâce à l'interface utilisateur. Toute application d'entreprise intégrant un environnement client-serveur doit être munie d'une interface utilisateur, techniquement appelée Graphical User Interface (GUI). Elle dispose des éléments d'affichage permettant de montrer les réponses aux requêtes émises par l'utilisateur.

1.2.1.2 Applications

Ce module de l'architecture renferme l'ensemble de tous les traitements effectués par le système. Cela englobe l'analyse, le traitement et l'exécution des requêtes utilisateur. La répartition des tâches entre le client et le serveur concernant le traitement diffère d'un type d'architecture C/S à l'autre. On peut rencontrer diverses possibilités:

- L'architecture un-tiers
- L'architecture deux-tiers
- L'architecture trois-tiers

1.2.1.3 Les données

Cette dernière partie de l'architecture a pour rôle de stocker les données de manière structurée et de les fournir à tout ce qui en demande. Les logiques de données, utilisées à la lecture, la création, la suppression, la mise à jour des données (CRUD, *Create Read Update Delete*), sont ses interfaces à tout système extérieur. En outre, cette opération, tout à fait indépendante du traitement, peut être gérée par un Système de Gestion de Base de Données (SGBD) comme Oracle, MySQL, SQL Server, ou autres.

1.2.1.4 Les middlewares

Les middlewares sont des entités intermédiaires qui assurent la communication entre les différentes entités de l'architecture, c'est-à-dire l'interface utilisateur-application et application-données. Cette communication est, en fait, assurée grâce à un protocole commun entre les deux entités. De surcroît, le middleware assure aussi la présentation de ces données dans les formats appropriés à chaque entité pour permettre un dialogue efficace entre eux.

1.2.2 Architecture un-tiers

Dans une architecture C/S un-tiers, la partie dédiée à l'interface utilisateur n'est présente que chez la machine cliente. De ce fait, tout le reste, (c'est-à-dire les applications, les données et même la logique de présentation) est centralisé dans le serveur. Le client, dépourvu de toute opération de traitement, est donc appelé client passif. C'est le cas qu'on rencontre dans les *MainFrames* qui sont des gros ordinateurs dotés d'une performance assez élevée et qui effectuent le stockage et le traitement des données à eux seuls.

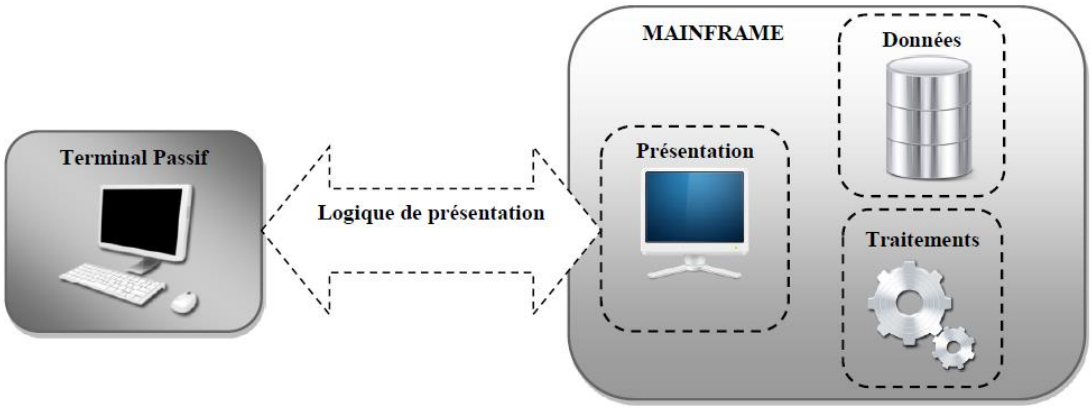

Figure 1.03: *Architecture Un-Tiers*

1.2.3 Architecture deux-tiers

C'est une architecture C/S de première génération dans laquelle on trouve la partie stockage de données centralisée sur un serveur. Le serveur n'est donc constitué que d'un serveur de données, tandis que le client se charge non seulement des opérations de traitement mais également de la présentation des données. Dès lors, le client est appelé client lourd, et les seules communications entre lui et le serveur consistent en l'envoi direct des requêtes de données au SGBD et l'acquisition des données retournées.

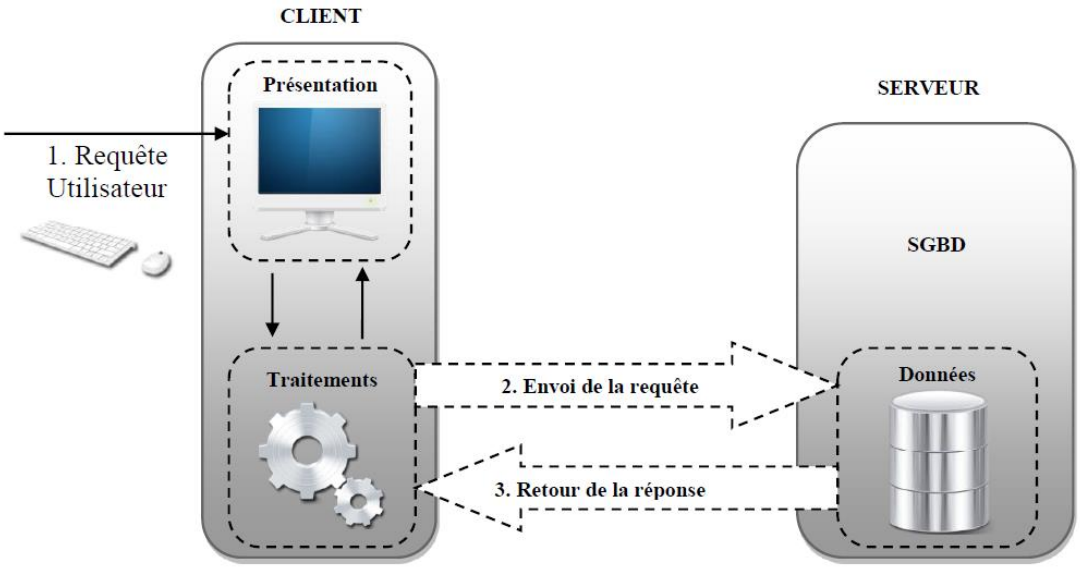

Figure 1.04: *Architecture deux-tiers*

1.2.4 Architecture trois-tiers

Dans ce type d'architecture, c'est la partie application qui est décomposée en deux subdivisions, telle que la plus grande est effectuée au niveau du serveur, notamment dans un serveur d'applications. Tous les logiques métiers y sont déployés, et c'est au serveur d'applications d'effectuer les requêtes et l'acquisition de données depuis le SGBD (ou autre système de stockage) du serveur de données. Seule une petite partie du traitement est effectuée au niveau du client.

Comme middleware, on rencontre le RPC (Remote Procedure Call) dans l'architecture trois-tiers. Ce dernier assure la communication entre le client, le serveur et les éventuels intermédiaires.

Contrairement à l'architecture C/S de données développée précédemment, cette architecture présente l'avantage d'être plus aisée à mettre à jour, car le logique métier centralisé au niveau du serveur peut être modifié sans aucune intervention nécessaire au niveau des clients.

On peut illustrer l'architecture trois-tiers comme suit :

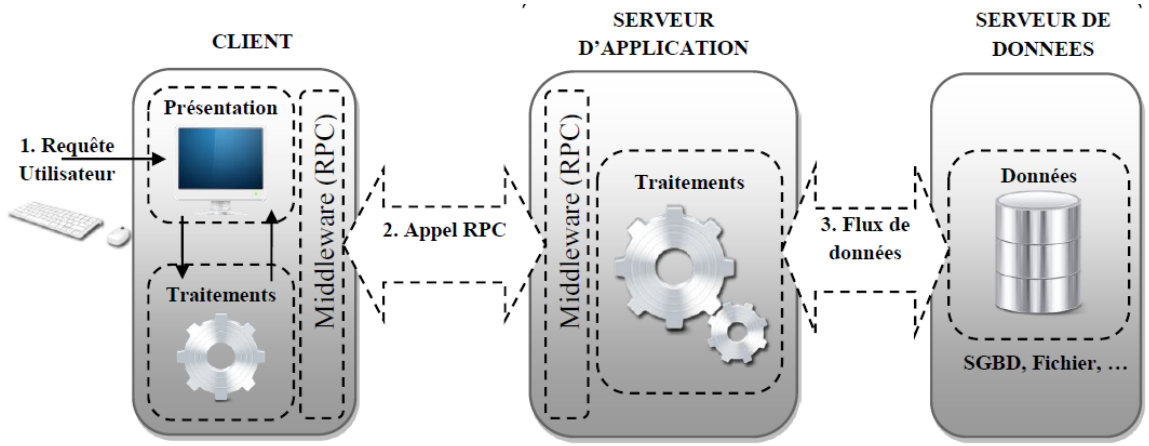

Figure 1.05: *Architecture trois-tiers*

1.2.5 Architecture n-tiers

L'architecture n-tiers ou multi-tiers est orientée vers la distribution d'application entre de multiples services. D'une manière explicite, une application est exécutée par plusieurs composants logiciels différents. Néanmoins, elle n'implique pas une mise en œuvre de plusieurs niveaux intermédiaires, mais une répartition des applications tout en gardant les trois niveaux de l'architecture distribuée.

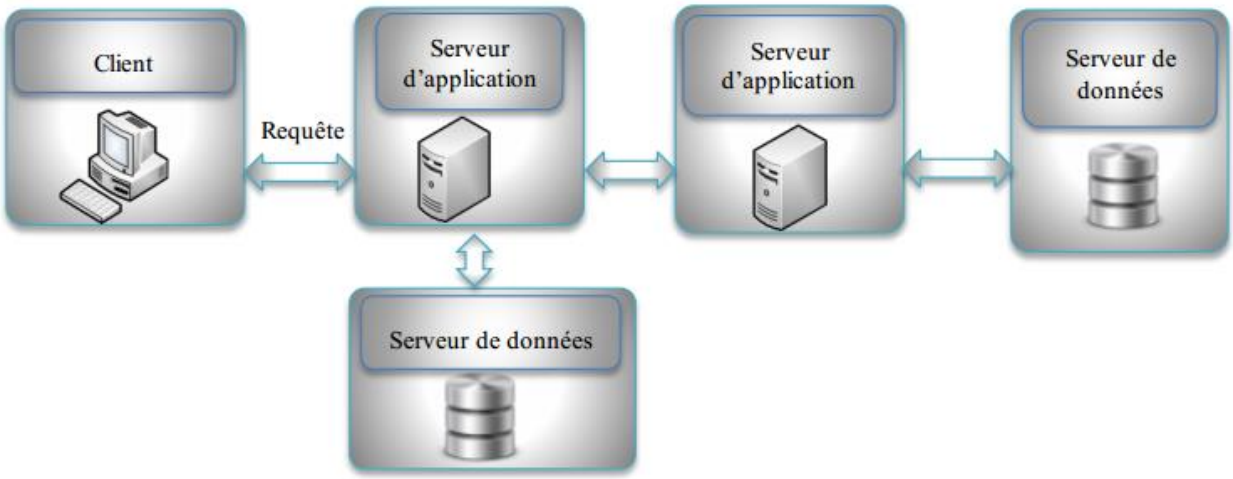

Figure 1.06: *Architecture n-tiers*

La distribution d'application est aperçue au niveau des serveurs intermédiaires, c'est-à-dire, une application (ou un service) se trouvant dans un serveur peut solliciter une ou plusieurs autres sur un ou plusieurs serveurs différents. Ces derniers peuvent en faire de même à leur tour, et ainsi de suite.

1.2.6 Avantages

Dans l'architecture C/S, toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels. Dans ce cas, il gère des ressources communes de tous les utilisateurs, par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction. La complexité du traitement et la puissance de calculs sont, en effet, à la charge du ou des serveurs, les utilisateurs utilisent seulement un client léger sur un ordinateur. Cette architecture, aussi flexible comme elle est, son évolution peut être assurée par l'ajout ou la suppression des clients ou serveurs sans perturber son fonctionnement fondamental.

1.2.7 Inconvénients

Certes, la découverte de l'architecture C/S a amené à beaucoup d'inventions technologiques pourtant elle présente quelques inconvénients. Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair-à-pair fonctionnent mieux en ajoutant de nouveaux participants). Dans le cas où le serveur ne serait plus disponible, les clients faisant référence à ce dernier resteront hors du système.

1.3 La technologie Web

1.3.1 Définitions

a). Internet

Internet est un réseau informatique mondial accessible au public. C'est un réseau de réseaux, sans centre névralgique, composé de millions de réseaux aussi bien publics que privés, universitaires, commerciaux et gouvernementaux. L'information est transmise par Internet grâce à un ensemble standardisé de protocoles de transfert de données, qui permet l'élaboration d'applications et de services variés comme le courrier électronique, la messagerie instantanée, le partage de fichiers en pair à pair et le World Wide Web[3].

b). Le Web

Le World Wide Web, communément appelé le Web, est un système hypertexte public fonctionnant sur Internet qui permet de consulter avec un navigateur des pages mises en ligne dans des sites.

1.3.2 Historique du Web

Le Web1.0 encore appelé Web statique était perçu dans les années 90 comme un ensemble de pages statiques symbolisant le transfert de la communication papier sous forme numérique par des pages HTML. A la suite de ce dernier est né un nouveau concept, concept proposé par Tim O'Reilly en 2004, mais consolidé en 2005 afin d'être baptisé Web2.0. Par ailleurs, cette nouvelle vision considère l'internaute comme acteur primaire qui contribue à la production, la consultation et la diffusion des contenus des sites: ce qui lui a valu d'être un Web participatif et social.

Depuis 2008, on a parlé d'une nouvelle vision du Web qui porte les noms de Web3.0 (encore appelé Web sémantique) et Web3D caractérisés par leur mobilité, portabilité, accessibilité et leur sens collaboratif. C'est une solution qui permet de faire interagir plusieurs acteurs sur un même document. Le Web4.0, quant à lui, est la vision du Web la plus récente, mais encore en cours de développement. Pour Nova Spivack, ce sera l'imposition du WebOS (Web Operating System) ou système d'exploitation Web qui n'est rien qu'un système d'exploitation omniprésent, intelligent et qui s'adapte aux habitudes des internautes pour d'autres. On aura donc la possibilité de travailler avec des outils uniquement en ligne ; et pour Joël de Rosnay ou Seth Godin, ce sera un Web symbiotique.

Statut	Appellation	Précision
Déjà déployé	Web 1.0	Web statique
	Web 1.5	Web dynamique
	Web 2.0	Web participatif
	Web 2.5	Web transformé en plate-forme pour les applications en ligne
	Web 2.B	Web 2.0 orienté pour le commerce
	Web² (Squared)	étape intermédiaire entre le Web 2.0 et le Web 3.0.
	Web 3.0	Web par P2P (Peer 2 Peer) d'un ordinateur à l'autre sans serveur
	Web3d	sites Internet 3D
Développement en cours	Web 4.0	Web symbiotique

Tableau 1.01: *Les évolutions du web*

1.3.3 Les différents types de ressources du Web

Les divers types de ressource du Web ont des usages assez distincts :

- les ressources constituant les pages web : documents HTML, images, scripts, feuilles de style, sons et vidéos;
- les ressources accessibles depuis une page web, mais consultables avec une interface particulière : applet;
- les ressources conçues pour être consultées séparément : documents (Word, PDF...), fichier texte, images de tout type, morceaux de musique, vidéo et d'autres fichiers à sauvegarder.

1.3.4 Conception

1.3.4.1 Universalité

Le Web a été conçu pour être accessible avec les équipements informatiques les plus divers : station de travail, terminal informatique en mode texte, ordinateur personnel, téléphone portable, etc. Cette universalité d'accès dépend en premier lieu de l'universalité des protocoles Internet. En second lieu, elle dépend de la flexibilité de présentation des pages web, offerte par HTML. En outre, HTTP offre aux navigateurs la possibilité de négocier le type de chaque ressource. Enfin, CSS permet de proposer différentes présentations, sélectionnées pour leur adéquation avec l'équipement utilisé.

1.3.4.2 Décentralisation

Les technologies du Web n'imposent pas d'organisation entre les sites web. Toute page du Web peut contenir des hyperliens vers toute autre ressource accessible d'Internet. Entre autres, il n'y a pas de registre centralisé d'hyperliens, de pages ou de sites. Le seul registre utilisé est celui du DNS. C'est une base de données distribuée qui répertorie les hôtes, et qui permet de traduire en adresse IP le nom de domaine contenu dans certains hyperliens.

1.3.5 Avantages

Cette évolution permanente du Web s'accompagne de nombreux avantages avec notamment une meilleure qualité de l'information, une augmentation du nombre d'internautes sur la toile, un élargissement de la population d'internautes, une utilisation optimale de l'information en supprimant les redondances et en lui ajoutant une précision remarquable au niveau de la recherche, un meilleur traitement des articles et une optimisation du temps, de l'effort et du coût.

L’architecture du Web favorise aussi les recherches thématiques (répertoires) et par mots-clés (moteurs de recherche). Grâce à des moteurs de recherche (Google, Yahoo…), on peut rechercher à partir de plusieurs mots-clés en une seule opération.

1.3.6 Inconvénients

Face au nombre d’internautes et serveurs, on peut dire que le contenu du Web n'est pas toujours fiable et il est même parfois médiocre. De plus, beaucoup de pages Web ne sont pas à jour, il y a des pages supprimées et adresses modifiées sans préavis.

1.4 Architecture Orientée Service (SOA)

1.4.1 Définition

Une SOA est une méthodologie de conception des systèmes informatiques organisationnels basée sur la notion de service. Ainsi, quand on parle de système SOA, on se rend à l’idée qu’il y aura un « Accès à un service ».A part le service, il y a aussi l’annuaire, le contrat, les fournisseurs et clients. Tous ces derniers forment le noyau et le point central de la SOA.

Les rôles des éléments cités précédemment sont illustrés sur la figure suivante.

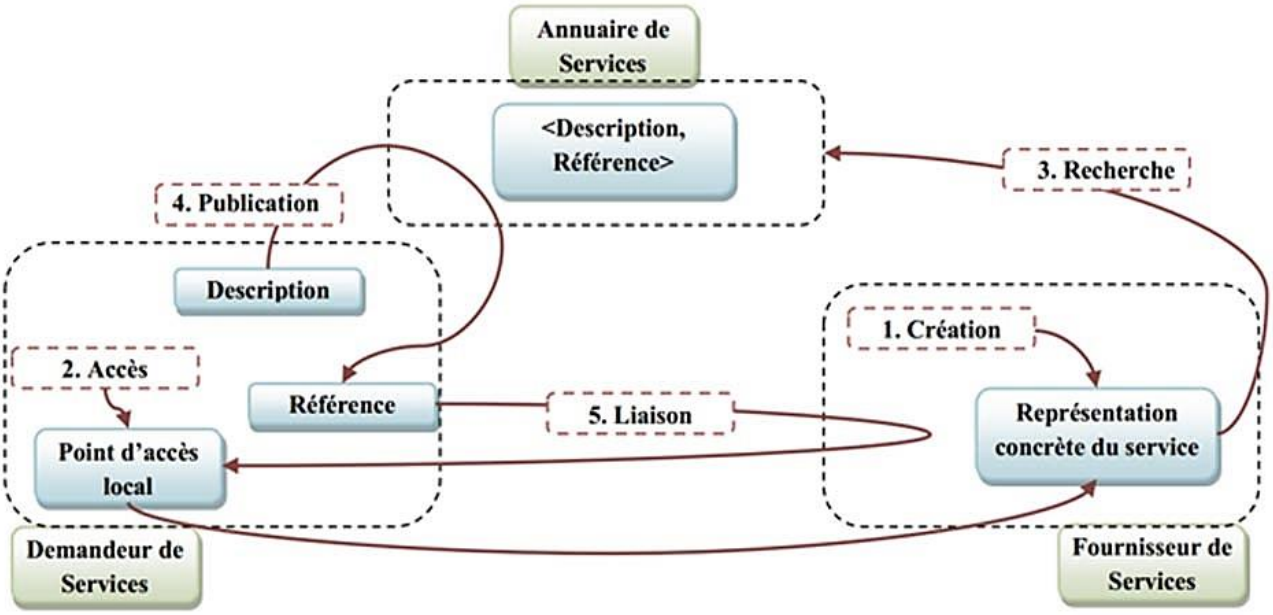

Figure 1.07: *Accès à un service pour une SOA [4]*

1.4.2 Les éléments dans une SOA

a). Le service

Un service est, à la base, un ensemble de fonctionnalités accomplissant des tâches spécifiques, accessible à travers un réseau. La définition des tâches qu'un service peut effectuer ainsi que la structure des données échangées constituent le contrat de service. L'application qui fournit un service est appelée «fournisseur ou prestataire de service» et l'application qui utilise le service «client». Cependant, une application peut jouer à la fois le rôle de fournisseur de service et le rôle de client de service.

Du point de vue commercial, un service peut être gratuit s'il ne demande aucune forme de rémunération ni d'échange; ou payant s'il requiert une sorte de rémunération avec un mode de paiement bien défini. Dans ce cas, des pénalités pourront être définies en cas de non satisfaction des termes du contrat. Un service est dit troqué s'il peut être fourni en échange d'un autre; et il est dit mixte s'il est à la fois troqué et payant, c'est-à-dire que la rémunération peut se présenter à la fois sous forme d'échange et de numéraire.

b). Annuaire de service

L'annuaire de services fait référence à l'ensemble des services (et des contrats associés) disponibles au sein du Système d'Information (SI). Il participe ainsi activement à la mise en œuvre d'une cartographie dynamique du SI. Dans un modèle de bus, l'annuaire peut être autoalimenté par le service (enregistrement). Les annuaires UDDI (*Universal Description Discovery and Integration*) forment aujourd'hui le standard de référencement des services.

c). Contrat de service

L'accomplissement d'un service par un fournisseur à la demande d'un client est régi dans un contrat de service. Le contrat de service est spécifié par le fournisseur de service. Il s'agit d'un document qui définit:

- Les fonctions du service qui décrivent de manière abstraite l'offre de service ;
- L'interface du service qui décrit les mécanismes et les protocoles de communication avec le prestataire de service ;
- La qualité du service qui désigne les détails de fiabilité, de disponibilité, de robustesse, etc.

d). Les protocoles et les normes

Une SOA se repose principalement sur l'utilisation d'une interface et de vocabulaire de description de données qui doivent être communes à l'ensemble des agents (fournisseurs de services et utilisateurs de services).

Parmi les différentes couches de normes et protocoles qui permettent de bâtir de telles architectures, on relève :

- la **gestion d'un annuaire** de services: UDDI (*Universal Description Discovery and Integration*) normalisé par l'OASIS(*Organization for the Advancement of Structured Information Standards*),
- la **description des interfaces** des services: WSDL (*Services Description Language),* recommandé par le W3C (*World Wide Web Consortium*),
- l'**invocation** (ou l'appel) du service: SOAP (*Simple Object Access Protocol*), recommandé par le W3C,
- le **format des données** échangées: XML (*eXtensible Markup Language*), recommandé par le W3C,
- le **transport des données** avec les protocoles internet : HTTP (*HyperText Transfer Protocol*) et TCP/IP (*Transmission Control Protocol/ Internet Protocol*), recommandations RFC (*Requests For Comments*).

1.4.3 Avantages

Une SOA a comme avantage de favoriser la réutilisation de composants logiciels d'une part, et de faciliter l'intégration de systèmes hétérogènes d'autre part. De ce fait, les efforts et les coûts de développement sont réduits. Dans un contexte de mobilité, la solution qu'ils apportent face aux problèmes d'interopérabilité est essentielle: toute plateforme mobile, peu importe le matériel ou le système d'exploitation, a la capacité d'interagir de manière uniforme dans le système à une structure orientée service. Leur support de communication asynchrone les rend également bien adaptés aux réseaux sans-fil qui ne peuvent garantir des connexions persistantes: chaque requête étant indépendante l'une de l'autre, le client peut contacter le service au moment opportun, lorsque la connectivité réseau est disponible. Il n'est pas nécessaire d'établir une connexion persistante et de gérer des sessions, contrairement à ce qui se fait dans les architectures C/S traditionnelles.

Un autre point important concernant la répartition des rôles dans une SOA est qu'il devient possible de déléguer à un service les traitements plus complexes (consommateurs de temps processeur), au lieu de les réaliser à même l'appareil mobile. De cette manière, un puissant serveur qui héberge le service arrivera à exécuter ce traitement en moins de temps que l'appareil mobile, qui comporte un processeur plus lent et une mémoire vive limitée.

1.4.4 Inconvénients

L'architecture SOA présente un coût de conception et de développement initiaux élevé. De plus, la disponibilité des services nécessite une duplication des données sur plusieurs serveurs qui affaiblit la sécurité. Faute des couches supplémentaires, la performance est aussi réduite pour des traitements simples. Par ailleurs, la tolérance à la panne est faible.

1.5 Système GRID

1.5.1 Définition

Une grille informatique ou Grid Computing est souvent confondue avec le Cloud Computing. En effet, le Grid Computing est l'utilisation des ressources de plusieurs machines sur le réseau pour traiter un problème à un même instant donné. C'est donc une infrastructure virtuelle constituée d'un ensemble de ressources informatiques potentiellement partagées, distribuées, hétérogènes, délocalisées et autonomes[5].

Une grille est avant tout une infrastructure, c'est-à-dire des équipements techniques d'ordre matériel et logiciel. Cette infrastructure est qualifiée de virtuelle car les relations entre les entités qui la composent n'existent pas sur le plan matériel, mais d'un point de vue logique.

Certes, une grille se compose de ressources informatiques: tout élément qui permet l'exécution d'une tâche ou le stockage d'une donnée numérique. Cette définition inclut effectivement les ordinateurs personnels ainsi que les téléphones portables.

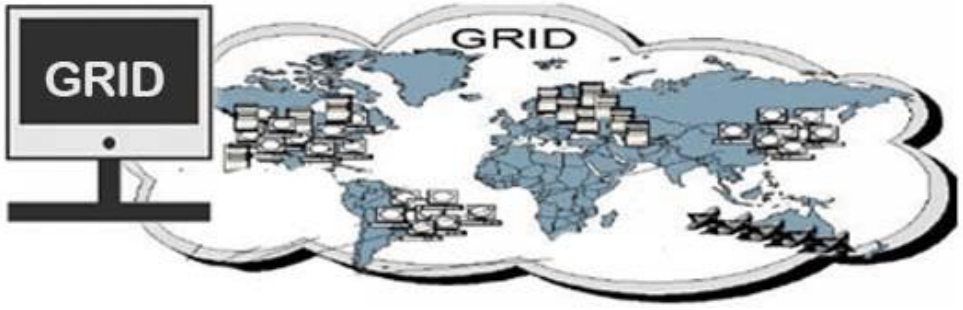

Figure 1.08: *Modélisation pratique du système Grid*

1.5.2 Principes de base du Grid Computing

Une grille ou grid est toujours basée sur un principe d'échanges C/S. Les serveurs fournissent des tâches à réaliser ou des données à traiter aux clients. Ils servent alors d'ordonnanceurs afin d'organiser le traitement. L'agrégation des retours des clients permet la création d'un résultat final. Les clients, proposent leur puissance de calcul ou de stockage à la grille afin de créer une sorte de supercalculateur.

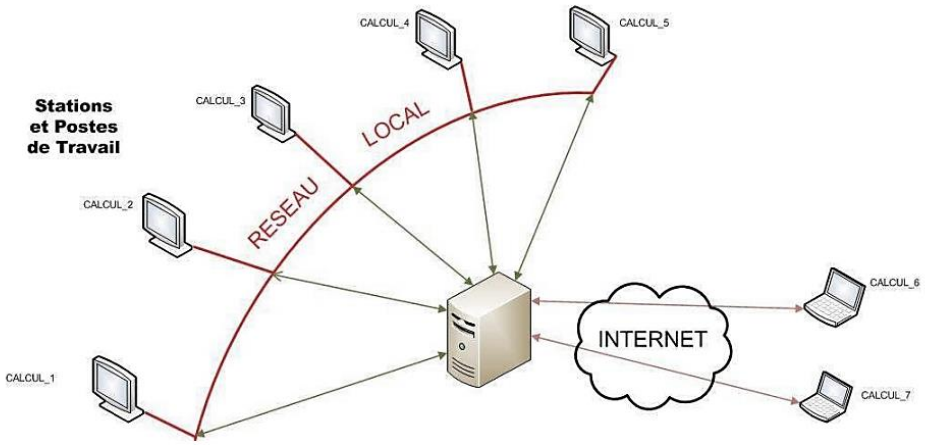

Figure 1.09: *Principe de base du système Grid[5]*

1.5.3 Les différents types de Grid Computing

Les terminologies utilisées dans les grilles sont mieux explicites dans leur version anglaise. C'est pourquoi plusieurs termes utilisés dans ce document seront gardés dans cette version. On essaiera néanmoins de leur donner une traduction contextuelle. Par ailleurs, on peut classer les grilles (*grid*) dans l'une des catégories suivantes.

1.5.3.1 Grille de calcul

Plusieurs ressources d'exécution hétérogènes sont reliées pour fusionner leur puissance et obtenir une performance de calcul élevée. L'unité de mesure de la performance d'un tel système est le FLOPS ou *FLoating-point Operations Per Second*, qui correspond au nombre d'opérations sur des réels (*floating-point*) par cycle d'horloge. L'opération en virgule flottante peut ainsi être calculée à partir de différentes tailles de nombres : en simple précision (32 bits) ou en double précision (64 bits). Toutefois, les processeurs usuels des PC peuvent effectuer 4 FLOP par cycle. Un processeur à simple cœur de fréquence 2,5 GHz est donc équivalent à une performance de 10 GigaFLOPS.

Plusieurs projets de mise en place de ces types de grille sont déjà fonctionnels. Parmi d'autres :

- Le projet BOINC (*Berkeley Open Infrastructure for Network Computing*) est une plateforme qui rassemble des millions d'ordinateurs volontaires connectés à internet pour participer au traitement intensif dans différents domaines tel que la recherche extraterrestre (SETI@home ou *Search for ExtraTerrestial Inteligence*), qui possède une performance d'environ 600 TeraFLOPS. La grille Einstein@home, vouée à la recherche astrophysique, qui repose sur la même plateforme (BOINC), possède une puissance de 900 TeraFLOPS.
- Le projet EGEE (*Enabling Grid for E-sciencE*) est une initiative de la commission européenne pour coupler plusieurs grilles de calculs déjà existants dans le but de fournir aux scientifiques une puissance de calcul et de stockage élevée, afin de lancer des simulations massives. Ainsi, pour eux, il s'agit d'une solution moins couteuse que l'emploi des supercalculateurs.

1.5.3.2 Data grid (Grille de stockage)

Les grilles de stockage permettent à leurs utilisateurs de conserver de très grands volumes d'informations mesurés en PetaOctetcs. Ces types de grilles sont le plus souvent accompagnés de grilles de calcul. TeraGrid en est un exemple, qui unifie des centres de supercalculateurs aux Etats-Unis pour fournir un total de 0.6 PetaOctets de capacité de stockage, 13 TeraFLOPS de puissance de calcul et plusieurs dizaines de TeraOctets de mémoire vive.

1.5.4 Avantages

Le système Grid rend possible l'exploitation d'un parc informatique à sa pleine puissance. Sans gaspillage, la puissance inutilisée sera répartie sur d'autres applications, souvent distantes. La collaboration entre les machines géographiquement très éloignées est aussi possible, grâce à cette technologie. Dès lors, la réalisation des calculs qui prenait beaucoup de temps peut se faire en un laps de temps acceptable. L'assemblage de plusieurs éléments réduit le coût financier par rapport à l'achat d'un supercalculateur. Un des avantages d'une grille informatique est aussi sa flexibilité. Il est possible d'ajouter ou supprimer un ou plusieurs composants du système. De ce fait, la tolérance aux pannes est élevée.

1.5.5 Inconvénients

Derrière l'idée de mutualiser les temps processeurs inutilisés pour exécuter des calculs distribués se cache tout de même un certain nombre d'inconvénients et de contraintes. On rencontre que les

applications doivent être qualifiées pour tourner sur une telle architecture. De plus, il n'y a pas encore de standard de développement d'application pour le système Grid. Pour les applications développées, la diversité des méthodes de travail pour le lancement des calculs rend les utilisateurs mal à l'aise. Par ailleurs, le faible niveau de sécurité dissuade les entreprises de se migrer vers le système Grid.

1.6 Virtualisation

1.6.1 Définition

La définition "formelle" de la virtualisation fait référence à l'abstraction physique des ressources informatiques. En d'autres termes, les ressources physiques allouées à une machine virtuelle sont abstraites à partir de leurs équivalents physiques. Les disques virtuels, interfaces réseau virtuelles, réseaux locaux virtuels, commutateurs virtuels, processeurs virtuels et la mémoire virtuelle correspondent tous à des ressources physiques sur des systèmes informatiques physiques. En fait, l'ordinateur hôte voit ses machines virtuelles comme des applications auxquelles il dédie ou distribue ses ressources [6].

Il existe de nombreux types de virtualisation: d'application, de plateforme, de réseau et des données. Généralement, quand quelqu'un mentionne la virtualisation, il fait référence à la virtualisation de plate-forme. Celle-ci correspond à l'utilisation de matériel serveur pour héberger plusieurs machines virtuelles invitées. Chaque machine virtuelle est un environnement virtuel sur lequel est installé un système d'exploitation. Elles fonctionnent indépendamment des autres.

1.6.2 Les techniques utilisées en virtualisation

1.6.2.1 L'isolation

L'isolation vise à isoler chaque processus dans un conteneur dont il est théoriquement impossible d'en sortir. Un processus isolé de la sorte ne saura pas quels autres processus s'exécutent sur le même système, et n'aura qu'une vision limitée de son environnement. D'où, cette isolation peut-être plus ou moins complète. L'exemple le plus connu est l'isolation du répertoire racine. L'application isolée voit une racine de son système de fichier qui est en fait un sous-répertoire du système de fichier complet. Le but principal de cette technologie est d'améliorer la sécurité du système d'exploitation et des applications.

1.6.2.2 L'émulation

C'est la capacité d'un logiciel (émulateur) à exécuter un système d'exploitation ou une application en lui présentant un matériel qui n'est pas le matériel réel. Avec un émulateur "pur", le code du système d'exploitation ou de l'application est exécuté et est complètement interprété par l'émulateur.

1.6.3 Les différents types de virtualisation

1.6.3.1 Virtualisation d'application

La virtualisation d'application est un terme générique pour désigner le fait de séparer l'utilisation d'une application des environnements matériels et logiciels nécessaires à son exécution. Elle consiste à télécharger une application sur le poste de travail de façon à l'exécuter sans avoir à l'installer au sens habituel du terme (streaming d'application). Il s'agit d'une technologie qui présente l'intérêt de simplifier radicalement le déploiement d'applications. Il suffit en effet de maintenir une image sur le serveur pour mettre à jour tous les postes clients. De plus, chaque application s'exécutant dans ce mode est isolée des autres ce qui résout toute question liée à la compatibilité applicative. Enfin, cette solution fonctionne en mode connecté et déconnecté.

1.6.3.2 Virtualisation des données

La virtualisation des données est une représentation virtuelle des données au lieu d'une représentation physique. Elle consiste à rediriger automatiquement les stockages des données à un utilisateur connecté depuis n'importe quel poste du réseau tout en conservant ses paramètres personnalisés. D'ailleurs, ce type de virtualisation aboutit à une grande flexibilité en permettant à chaque collaborateur de travailler depuis n'importe quel emplacement.

1.6.3.3 Virtualisation de la présentation

La virtualisation de présentation consiste à exécuter des applications de manière centralisée sur des serveurs de présentations et à en déporter l'affichage sur le poste client par un protocole spécifique. Cette technologie permet la centralisation des applications et des données en assurant une sécurité optimale. Citrix XenApp et Microsoft Remote Desktop Services sont les deux technologies principales des environnements Windows. Ulteo, une solution Open Source, propose un accès transparent aux applications Windows et Linux.

1.6.3.4 Virtualisation de réseau

La virtualisation du réseau consiste à combiner des ressources réseau matérielles et logicielles dans une seule unité administrative. Elle permet de résoudre un certain nombre de problèmes liés aux stockages, à la capacité de calcul, à la consommation d'énergie et à l'accessibilité aux données en permanence.

1.6.3.5 Virtualisation de plateforme

a). La virtualisation complète

La virtualisation complète (full virtualization) consiste à émuler l'intégralité d'une machine physique pour le système invité. Le système invité « croit » s'exécuter sur une véritable machine physique. Le logiciel chargé d'émuler cette machine s'appelle une machine virtuelle. Son rôle est de transformer les instructions du système invité en instructions pour le système hôte.

b). La paravirtualisation

La paravirtualisation est très proche du concept de la virtualisation complète, dans le sens où c'est toujours un système d'exploitation complet qui s'exécute sur le matériel émulé par une machine virtuelle. Cette dernière s'exécute au-dessus d'un système hôte. Toutefois, dans une solution de paravirtualisation, le système invité est modifié pour être exécuté par la machine virtuelle. Les modifications effectuées visent à rendre le système émulé « au courant » du fait qu'il s'exécute dans une machine virtuelle. De ce fait, il pourra collaborer plus étroitement avec le système hôte, en utilisant une interface spécifique, au lieu d'accéder au matériel virtuel via les couches d'abstraction.

La paravirtualisation apporte un gain de performances avéré du fait du contournement des couches d'abstraction. En effet, comme le système invité collabore activement avec la machine virtuelle, il ne se comporte plus comme un système d'exploitation à part entière s'exécutant directement sur du matériel. Au contraire, il adapte son comportement pour que les accès au matériel, souvent difficiles à interpréter de manière efficace par la machine virtuelle, soit transformés en des appels directs à cette dernière.

a). La virtualisation assistée par le matériel ou à hyperviseur

L'utilisation d'un hyperviseur est l'évolution logique de la paravirtualisation. Certes, un hyperviseur est un noyau allégé qui s'intercale entre le matériel et les systèmes invités. Dans les

technologies précédentes, le système hôte était le seul à avoir un accès direct au matériel; avec un hyperviseur, le système hôte partage cet accès avec les systèmes invités. Si les deux technologies vues précédemment (virtualisation complète et paravirtualisation) utilisaient une machine virtuelle pour émuler le matériel, il n'en va pas de même avec un hyperviseur. Chaque système d'exploitation a un accès presque direct au matériel par l'intermédiaire de l'hyperviseur.

Avec un hyperviseur, le contrôle de l'accès au matériel et de l'utilisation des ressources est bien plus fin. En effet, dans les solutions à base de machine virtuelle, le système invité est vu comme un processus par le système hôte. Le niveau de contrôle et de suivi est donc celui offert par le système hôte. Ici, c'est l'hyperviseur qui est chargé d'appliquer la politique d'accès aux ressources matérielles.

1.6.4 Avantages

Le taux d'utilisation moyen d'un serveur tourne autour des 20 %. Une infrastructure virtualisée permet en effet de rentabiliser le matériel en répartissant efficacement plusieurs machines sur les machines physiques. En plus de mieux utiliser l'existant, mutualiser son environnement peut engendrer une baisse significative des coûts associés. La baisse de la consommation énergétique et du besoin en refroidissement par exemple peut être une source importante d'économies. La virtualisation diminue aussi le nombre de périphériques et de pilotes à gérer.

Les données d'un OS virtuel peuvent être sauvegardées. Cette possibilité présente l'avantage de pouvoir « promener » son système n'importe où, et celui de pouvoir le restaurer facilement en cas de besoin. En outre, les machines virtuelles représentent un parfait environnement de test si bien que les développeurs pourront essayer leurs applications sans crainte de planter l'infrastructure physique.

1.6.5 Inconvénients

Malgré ces bénéfices, il reste néanmoins primordial de ne pas perdre de vue certains inconvénients inhérents à tout environnement virtuel. Le premier étant qu'en cas de panne de la machine physique, c'est l'intégralité des espaces virtuels qu'elle contenait qui est affectée. Enfin, certaines politiques de confidentialité peuvent ne pas être compatibles avec un stockage de données sur un serveur partagé.

1.7 Conclusion

En bref, internet favorise l'interconnexion des millions d'équipements informatiques. L'architecture C/S, Web, SOA et le système Grid exploitent ce réseau mondial comme support de transmission entre leurs éléments. La virtualisation donne une possibilité d'attribuer à une machine physique, plusieurs systèmes d'exploitation, applications et données indépendamment entre eux. On peut dire que le Cloud Computing a vu le jour grâce à l'existence de l'internet, quatre architectures de base (MainFrame, C/S, Web et SOA) et deux autres technologies (système Grid et virtualisation). Dans le chapitre suivant, nous allons voir ce qu'est réellement le Cloud Computing.

CHAPITRE 2

CONCEPT DU CLOUD COMPUTING

2.1 Introduction

Etant donné que le Cloud Computing devient de plus en plus important aux entreprises, le grand public commence à le connaître sous la forme de services de stockage à distance, ou encore de la musique en streaming. Malgré le fait que son nom a une forte confusion avec internet, qui est souvent représenté sous forme d'un nuage, peu de monde arrive tout de même à le définir.

Dans ce chapitre, nous allons voir dans les détails le Cloud Computing à partir de sa définition, son historique, ses caractéristiques, ses modèles de déploiement et ses différents types. A la suite, nous mettrons en évidence ses avantages et ses inconvénients.

2.2 Définitions

2.2.1 Cloud Computing

Comme le Cloud Computing ou « informatique dans les nuages » envahit le monde informatique, beaucoup d'ouvrages en parlent et possèdent leur propre définition. Néanmoins, ces différents ouvrages et la variété de leurs définitions contribuent à la bonne connaissance du sujet en question.

La première définition mentionne que le Cloud Computing est l'ensemble des disciplines, technologies et modèles commerciaux utilisés pour délivrer des capacités informatiques (logiciels, plateformes, matériels) comme un service à la demande [7].

De son côté, le bureau d'expertise technologique Wygwam de Microsoft définit le Cloud Computing comme un nouveau modèle informatique qui consiste à proposer les services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui [1].

Finalement, le NIST (North American National Institute for Standard and Technology) définit le Cloud Computing comme un modèle pour une mise à disposition simple et à la demande de ressources informatiques partagées et configurables (à l'exemple des réseaux, serveurs, systèmes de stockage, processeur, applications, etc).

2.2.2 Datacenter

Un datacenter est une infrastructure immobilière et technique permettant d'héberger des équipements informatiques (principalement des serveurs), et disposant de moyens propres à une exploitation performante (électricité, climatisation, accès haut débit) et sécurisée. Il est constitué d'un espace protégé et disposant d'une architecture généralement redondante (alimentation électrique, accès haut débit).

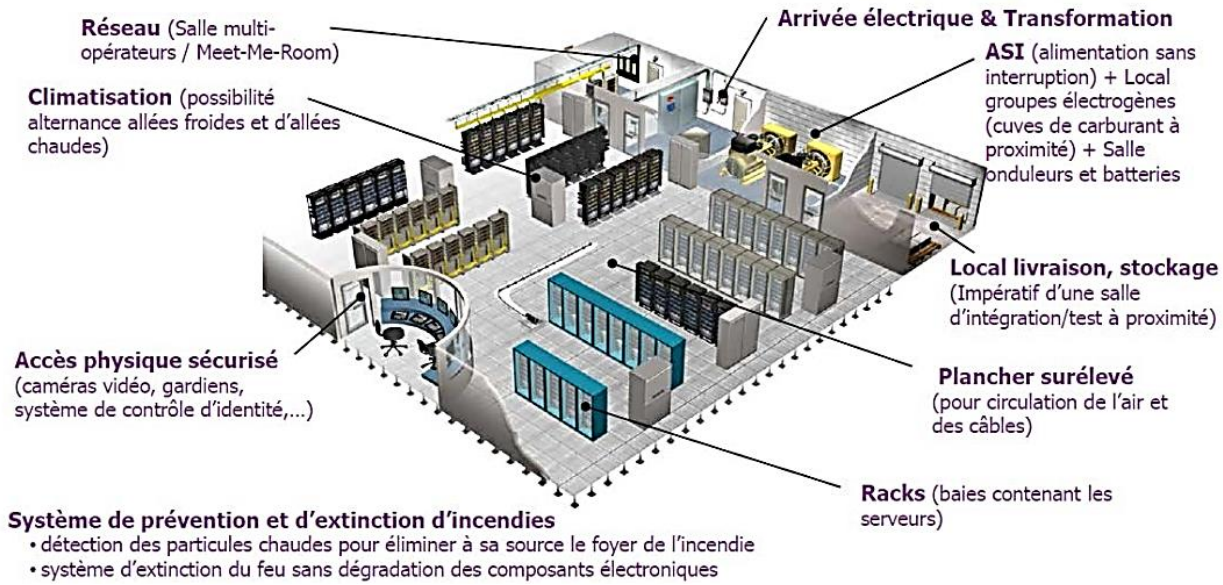

Figure 2.01: *Plan globale d'un datacenter*

2.3 Historique

Les principes sous-jacents au Cloud Computing remontent aux années 1950. À cette époque, les utilisateurs accédaient depuis leurs terminaux à des applications fonctionnant sur des systèmes centraux (*Mainframe*), qui correspondaient aux ancêtres des serveurs du Cloud.

Son histoire est aussi liée au développement d'Internet et des autres architectures informatiques, à savoir, l'architecture C/S, Web et SOA.

En juillet 2002, Amazon, un site de vente en ligne rencontrait une grande croissance. Pour faire face à la demande qui grandissait de jour en jour, il a dû installer des dizaines de milliers de serveurs dans le monde, eux-mêmes répartis dans de multiples datacenters: aux Etats-Unis, en Irlande, en Asie, etc. Tous ces serveurs n'étaient pas en fait, utilisés en même temps. Certains restaient prêts pour faire face à la demande lors des pics de vente (comme Noël). Les ingénieurs d'Amazon ont alors eu l'idée de louer les serveurs de garde à d'autres développeurs web. Ainsi,

les entreprises n'auront plus besoin d'acheter des tonnes de serveurs comme eux, juste pour être prêtes en cas de pic de trafic: ils devront simplement demander d'utiliser temporairement plus de serveurs les jours où ils ont plus de visiteurs.

De nos jours, le marché du Cloud Computing dépasse les 72 milliards de dollars. Selon les prévisions, d'ici 2020, cela devrait atteindre les 241 milliards de dollars [8].

2.4 Caractéristiques

2.4.1 Accès réseau universel

Un environnement Cloud est accessible via un réseau, quel que soit le périphérique (PC, Mac, tablette, Smartphone, etc). L'accès se fait généralement à l'aide d'un navigateur web, mais on rencontre aussi d'autres, via d'autres standards d'accès ou via les API fournis par les fournisseurs de service Cloud.

2.4.2 Mise en commun de ressources

Dans une architecture Cloud, le client ne pense pas en nombre de serveurs, taille de disques, nombre de processeurs… mais en puissance de calcul, capacité totale de stockage, bande passante disponible. Cela est dû à l'abstraction de certaines couches matérielles.

2.4.3 Service mesurable et facturable

Le fournisseur de service Cloud est capable de mesurer de façon précise la consommation des différentes ressources (CPU, stockage, bande passante, etc). Cette mesure lui permet de facturer les clients, qui est proportionnelle au temps d'utilisation et à la quantité de ressources qu'on lui a attribuées. De ce fait, il est possible à un utilisateur de consommer les services ou ressources sans pour autant nécessiter une demande d'intervention auprès du fournisseur.

2.4.4 Flexibilité

2.4.4.1 Cas d'un datacenter classique

Dans un datacenter classique, le taux moyen d'utilisation est en dessous de 25%. On retrouve aussi une barrière à l'innovation en raison de la nécessité d'un investissement initial, lourd en termes d'achats de matériels, de licences, etc. Le but est de saturer la charge serveur pour une exploitation optimale. Les serveurs sont sous-utilisés et en cas de pic de charge, il faut acheter de nouveaux serveurs pour satisfaire la demande [1].

Le schéma ci-dessous matérialise le niveau de charge d'une application sur l'axe du temps et des capacités IT mises à disposition.

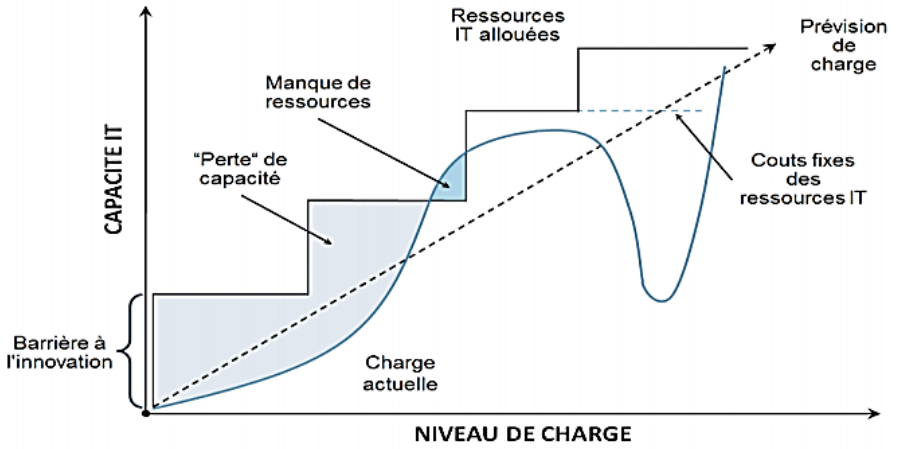

Figure 2.02: *Niveau de charge et capacité IT alloué dans un datacenter classique*

2.4.4.2 Cas d'un Cloud Computing

Étant donné que le Cloud Computing repose sur un ensemble de machines interconnectées et massivement distribuées, cela permet d'être tolérant face aux défaillances matérielles et logicielles. De plus, le Cloud Computing permet une flexibilité. Ainsi, la puissance de calcul ou les capacités de stockage peuvent être très facilement augmentées par l'installation de nouveaux équipements au sein ou en dehors du Cloud. La charge sera alors répartie en fonction de ces nouveaux équipements.

Certains fournisseurs proposent aussi un ensemble d'automatismes permettant d'ajuster dynamiquement la capacité IT en fonction du niveau de la charge. En cas de montée en charge, la création et le démarrage d'une nouvelle instance de serveur ne nécessitent que quelques minutes. Il en est de même pour la mise en arrêt d'une instance de serveur en cas d'une descente en charge.

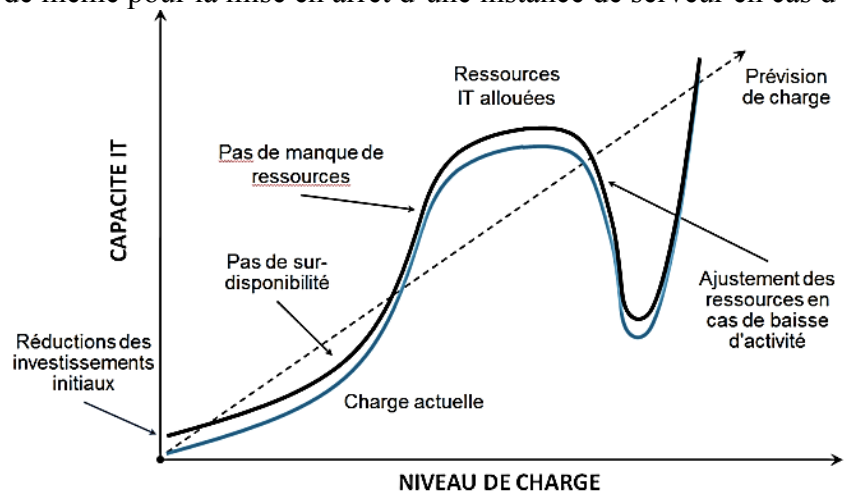

Figure 2.03: *Niveau de charge et capacité IT alloué dans un Cloud*

2.5 Modèles de déploiement

Tout d'abord, l'architecture générale du Cloud Computing se compose de trois éléments essentiels [9]:

- Les clients
- Le Datacenter
- Les répartitions de serveurs

Les clients sont les équipements que les utilisateurs finaux emploient pour avoir accès aux services offerts par le Cloud Computing. On peut avoir trois catégories de client:

- Client mobile: ce sont les appareils mobiles comme les smartphones et les tablettes
- Client pauvre: ce sont les ordinateurs qui n'ont pas de hardware interne, mais utilisent un serveur pour les traitements. Ils ne font que l'affichage des données.
- Client riche: ce sont les ordinateurs standards qui possèdent à la fois le hardware et le software pour accéder aux services du Cloud Computing.

Le Datacenter représente la collection de serveurs où les services fournis par le Cloud Computing sont hébergés. Il peut être une grande salle dans votre local ou une salle pleine de serveurs à l'autre bout du monde que le client accède via Internet. Les serveurs répartis sont les serveurs qui appartiennent au fournisseur de services Cloud, mais qui sont dispersés géographiquement à travers la planète.

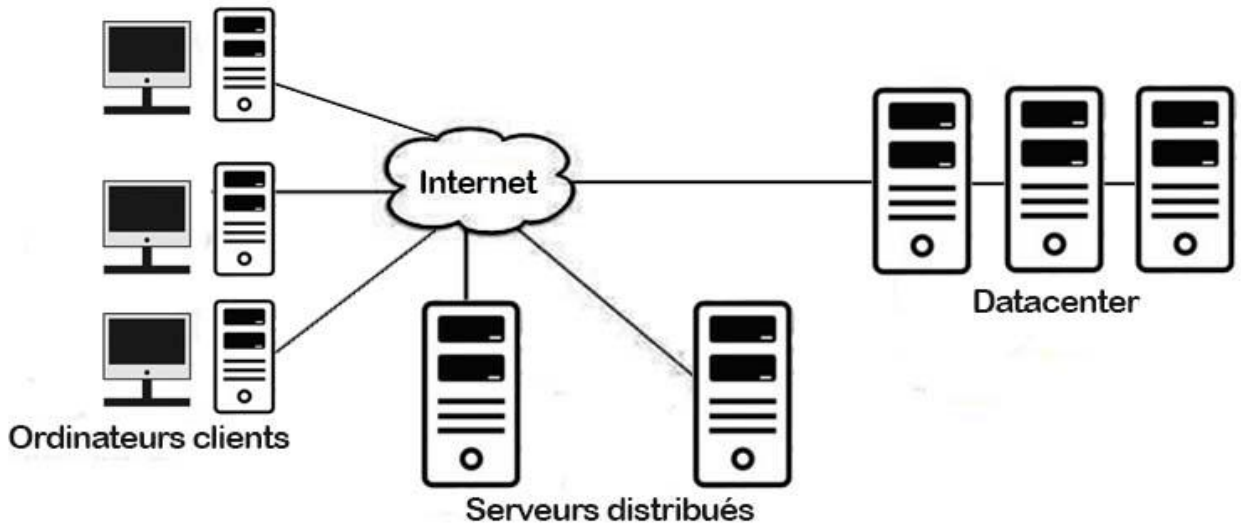

Figure 2.04: *Architecture générale du Cloud Computing*

Suivant l'architecture générale du Cloud Computing, on peut déployer le Cloud de quatre manières différentes:

- Le Cloud public
- Le Cloud privé
- Le Cloud hybride
- Le Cloud communautaire

2.5.1 Cloud privé

Un Cloud est privé si ses ressources physiques sont hébergées dans une infrastructure propre à l'entreprise et étant sous son contrôle. La maintenance et le déploiement des applications sont donc à sa charge.

Le Cloud privé peut aussi désigner un Cloud déployé sur une infrastructure physique dédiée et mise à la disposition d'un fournisseur de services. Ainsi, une entreprise peut louer à un fournisseur de services, un nombre conséquent de serveurs qui lui sont entièrement dédiés et sur lesquels une solution Cloud sera déployée pour gérer dynamiquement l'application, la plate-forme ou l'infrastructure (virtuelle).

2.5.2 Cloud publique

Le Cloud public est composé d'une multitude de services tiers accessibles depuis interne [7]. Il est dit public, car les ressources peuvent être partagées entre plusieurs entreprises. Pour ce modèle de déploiement, le fournisseur de la solution Cloud est externe. Il est propriétaire de son infrastructure et ses services sont accessibles à tout le monde. Certains services sont proposés par des fournisseurs géants, ce qui a pour conséquence d'avoir un éventail d'offre très important avec un coût unitaire potentiellement très faible. Amazon, Google et Microsoft proposent un Cloud public dans lequel n'importe quel particulier ou n'importe quelle entreprise peut y héberger ses applications, ses services ou ses données.

2.5.3 Cloud hybride

L'appellation «Cloud hybride» (ou Cloud mixte) est généralement utilisée pour désigner l'utilisation d'un Cloud privé avec celle d'un Cloud public. Plus simplement, on entend par Cloud hybride, l'utilisation de deux Cloud dans une organisation partageant les données et les applications [7]. Par exemple, un Cloud dédié uniquement aux données, et un autre aux applications

Le Cloud hybride est aussi une configuration associant des infrastructures privées, en mode Cloud ou non, et des Clouds publics ou privés. Dans ces cas-là, une partie des données ou des infrastructures est gérée en propre par l'entreprise, dans ses locaux ou chez un administrateur, et cette partie communique par ailleurs avec des ressources en Cloud. Ce type d'organisation permet de différencier le lieu de traitement selon différents critères : stratégiques, par exemple pour garder les données sensibles dans ses murs ; économiques, par exemple pour n'utiliser le Cloud que là où il est le plus rentable ; performance ; etc. Elle permet également d'utiliser uniquement le Cloud public en cas de débordement, c'est-à-dire, quand les capacités privées de l'entreprise sont dépassées, notamment lors d'un pic d'activité ou de trafic.

2.5.4 Cloud communautaire

Le Cloud communautaire est un modèle de Cloud Computing, dédié à une communauté spécifique, incluant partenaires et sous-traitants, pour travailler de manière collaborative sur un même projet. Les ressources, les services et la propriété sont partagés à l'échelle d'une communauté (par exemple à l'échelle d'un Etat, d'une ville, d'une académie,…).

Le schéma suivant résume les caractéristiques des 4 modèles de déploiement du Cloud qu'on a détaillé précédemment.

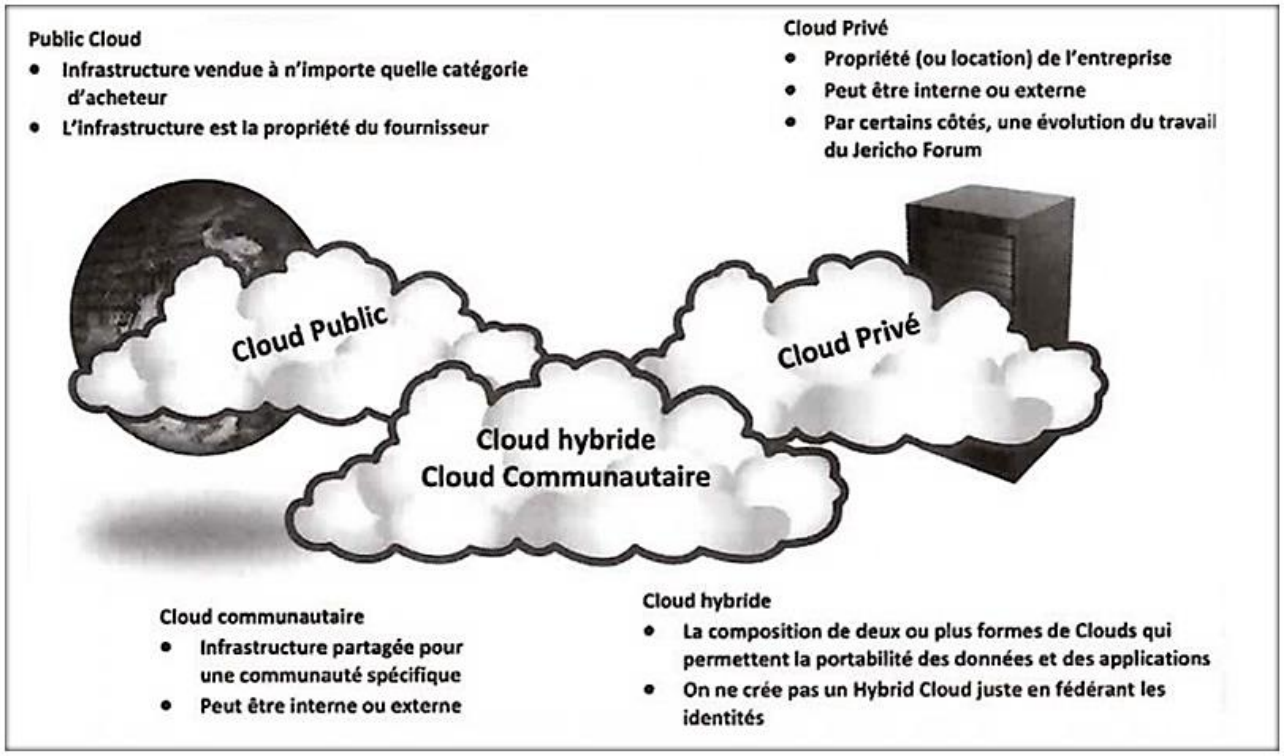

Figure 2.05: *Récapitulatif des modèles de déploiement Cloud[7]*

2.6 Différents types de service Cloud

Face aux types de services Cloud existants, le « Free Cloud Alliance », crée en mars 2010 a défini 3 types principaux: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) et SaaS (Software as a Service). Chacun de ces types de service peut être rencontré dans les 4 types de déploiement du Cloud (public, privé, hybride et communautaire).

2.6.1 Infrastructure as a Service : IaaS

2.6.1.1 Définition

L'IaaS ou « infrastructure comme un service », permet de disposer d'une infrastructure à la demande, pouvant héberger et exécuter des applications, des services ou encore stocker des données. Ces ressources sont surtout d'ordre physique tel que les serveurs, systèmes de stockage, commutateurs, routeurs et autres équipements [1].

2.6.1.2 Caractéristiques

Le fournisseur de service n'est responsable que du réseau, de l'espace de stockage, du matériel serveur et de la virtualisation, en d'autres termes des ressources physiques. Le client quant à lui est responsable du logiciel serveur, des bases de données, de l'intégration SOA, des environnements d'exécution (runtime) et des applications. De ce fait, il peut déployer et exécuter ses propres logiciels tels que des systèmes d'exploitation et des applications propriétaires. L'offre IaaS est utilisée, à titre d'exemple, pour l'hébergement de serveurs virtuels, la location de stockage et la location de capacité de calcul.

2.6.1.3 Exemples

Openstack d'Ubuntu, que nous allons aborder dans cet ouvrage, permet de créer un Cloud public ou privé sous forme d'IaaS. Son point fort réside sur sa forme de licence, distribuée selon les termes de licence Apache (licence de logiciel libre).

Il existe aussi d'autres options payant pour créer son propre Cloud et de fournir une IaaS. VMware propose vSphere pour transformer les matériels informatiques d'une entreprise en une plateforme partagée. En vérité, vSphere offre une possibilité de créer un «Datacenter virtuel» dans lequel on pourra héberger des machines virtuelles. Microsoft fait de même avec Hyper-V Cloud, qui est

aussi un moteur de virtualisation pour conter la suprématie de VMware. En somme, C'est un ensemble de solutions pour monter rapidement un Cloud privé en offre IaaS.

2.6.2 PaaS: Platform as a Service

2.6.2.1 Définition

Le PaaS ou « plateforme comme un service » implique principalement la mise à la disposition pour une entreprise d'environnement technique afin de développer des applications qui fonctionneront à distance [1]. Les applications déployées pourront alors consommer des ressources machines : processeurs, mémoires, disques.

2.6.2.2 Caractéristiques

Le client maintient uniquement ses applications tandis que le fournisseur du service Cloud maintient les runtimes, l'intégration SOA, les bases de données, le logiciel serveur, la virtualisation, le matériel serveur, le stockage et les réseaux, d'une manière explicite, le bon fonctionnement de la plateforme. Il existe aussi d'autres techniques à assurer tels que le basculement (fail-over) et la répartition de charge (load-balancing). Cela offre ainsi une grande flexibilité, et permet de tester rapidement un prototype ou encore d'assurer un service informatique sur une période de courte durée. Le PaaS favorise également la mobilité des utilisateurs puisque l'accès aux données et aux applications peut se faire à partir de n'importe quel périphérique connecté. Il fournit aussi plus que l'infrastructure, car il permet d'avoir un kit complet d'outils pour le développement d'application. En effet, le PaaS permet de gérer une application à partir de sa conception, son élaboration, son déploiement, jusqu'à la maintenance.

2.6.2.3 Exemples

Parmi les différentes offres de service rencontré sur le web, quelques fournisseurs de service Cloud domine le marché du PaaS. Ces principaux acteurs sont: Microsoft, Amazon, Google et Salesforce.

Windows Azure est la plateforme en nuage de Microsoft. C'est une offre d'hébergement d'applications et données aux entreprises. Amazon, la librairie en ligne, a profité quant à elle de son infrastructure informatique pour proposer EC2 (Elastic Computing Cloud), qui permet d'avoir à sa disposition des machines virtuelles. En outre, Google exploite ces serveurs repartis dans le

monde entier pour fournir une plateforme de conception et d'hébergement d'applications web. Cette plateforme est baptisé Google App Engine. De son côté, Salesforce a mis en place « Force.com » une plate-forme pour accueillir les applications d'entreprise.

2.6.3 SaaS: Software as a Service

2.6.3.1 Définition

Le SaaS ou « logiciel comme un service » s'agit de la mise à disposition d'un logiciel à la demande et à distance [1]. Ce logiciel n'est pas livré sous la forme d'un produit que le client installe en interne sur ses serveurs, mais en tant qu'application accessible à distance comme un service, sous forme d'un abonnement, par le biais d'Internet. Ainsi, l'achat de licences ne sera plus nécessaire pour les utilisateurs. Ces services sont accessibles directement sur les réseaux (publique ou privé), via un navigateur web, ou une API spécifique fournie par le fournisseur de service Cloud.

2.6.3.2 Caractéristiques

Le fournisseur gère l'ensemble allant de l'infrastructure physique jusqu'à l'exécution de l'application. Il assure le bon fonctionnement de l'application et la disponibilité de leur serveur. L'utilisation est alors transparente pour les utilisateurs, qui ne se soucient ni de la plateforme, ni de sa mise à jour, ni encore de son administration. Le client ne paye pas pour acquérir le logiciel, mais plutôt pour l'utiliser sur une durée déterminée. Le contrôle des données est partagé entre le client (qui crée et utilise les données) et le fournisseur de Cloud (qui héberge, sauvegarde et sécurise les données).

2.6.3.3 Exemples

Microsoft et Google prennent la plus grande partie du marché sous forme de SaaS. Ces deux géants informatiques proposent plusieurs séries d'applications, seulement pour un abonnement mensuel ou annuel. Chaque pack d'abonnement contient une suite d'application bureautique (traitement de texte, tableur, présentation…), un gestionnaire de messagerie et un gestionnaire de stockage de données.

Microsoft propose Office 365: une marque désignant les formules permettant d'acquérir la dernière version de Microsoft Office en un ensemble de services Cloud, en abonnement mensuel

ou annuel. A part cela, cet abonnement englobe aussi l'utilisation d'OneDrive, Exchange Online, Lync Online et SharePoint Online.

Google met à la disposition des clients Google Apps. C'est une offre qui inclut Gmail, Google Calendar, Google Talk et Google Docs.

2.6.4 Récapitulation graphique

L'architecture Cloud consiste à mettre à la disposition des clients des ressources informatiques, partant de la couche la plus basse (infrastructure) jusqu'à la couche la plus haute (application). Ce principe se regroupe dans trois types de services selon le « Free Cloud Alliance ».

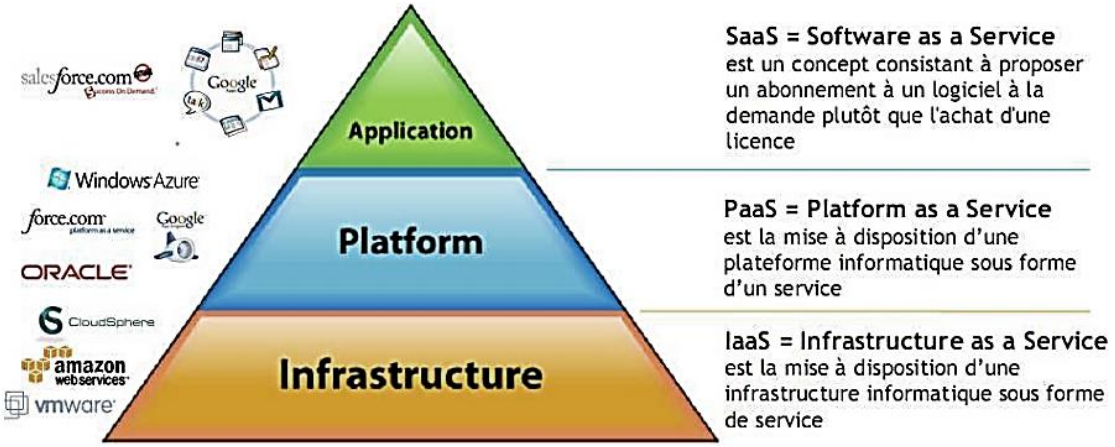

Figure 2.06: *Schéma global des types de service Cloud*

Chaque type de service Cloud définit précisément la responsabilité du client et du fournisseur de service. Cette responsabilité englobe neuf couches différentes, à partir de la couche réseau jusqu'à la couche application. Plus l'entreprise utilise la couche la plus haute du type de service Cloud (SaaS), plus sa responsabilité diminue.

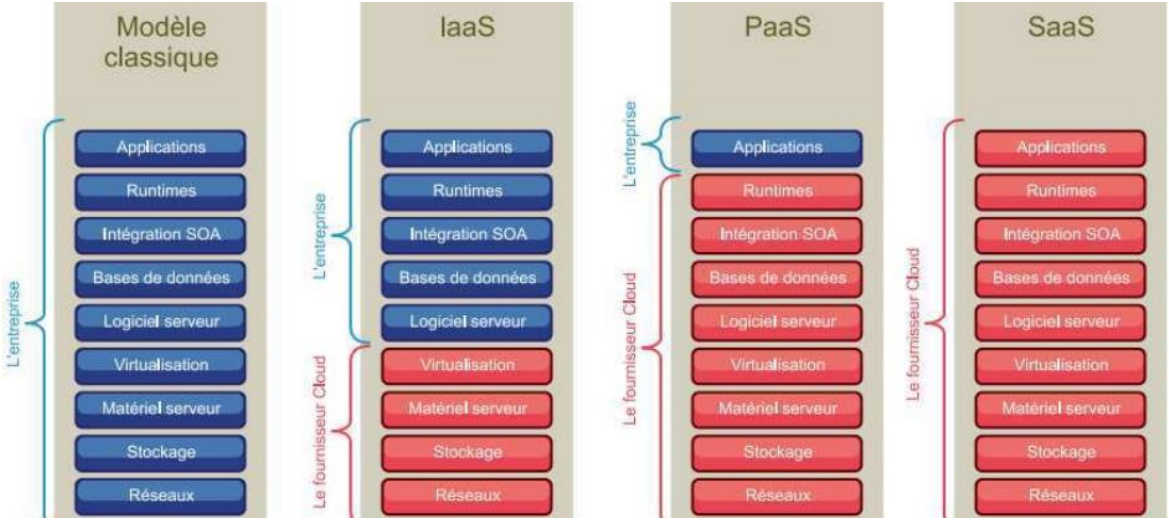

Figure 2.07: *Responsabilité du fournisseur selon le type de service Cloud*

2.7 Sécurité

Comme pour toutes les architectures informatiques, la sécurité est primordiale pour le Cloud Computing. Les risques liés à son utilisation doivent être pris en considération comparativement aux risques encourus par les environnements traditionnels.

2.7.1 Sécurité des données

2.7.1.1 Protection et récupération des données

Il existe deux métriques qui permettent de mesurer l'efficacité d'un processus de protection des données. Le premier est le *Recovery Time Objective* ou RTO, qui mesure le temps acceptable ou toléré de rétablissement du service lors d'une panne. Le second étant le *Recovery Point Objective* ou RPO, qui mesure la quantité de données que l'on tolère de perdre due à une panne ou au processus de restauration.

Une phase d'analyse est nécessaire pour identifier les applications et données critiques pour l'entreprise afin de mettre en place une politique de protection adaptée. Ainsi, plus les données sont critiques pour l'entreprise, plus la fréquence des copies doit être élevée avec un besoin d'accès et de performances plus important.

En cas de sinistre, il est aussi important de mettre en place une solution de miroir qui permet de protéger les données critiques. La réplication peut être synchrone ou asynchrone, en effectuant le cas échéant de la compression pour réduire le RPO. Dans ce cas, des mécanismes doivent être mis en place pour garantir la cohérence des données répliquées.

2.7.1.2 Intégrité des données

Les données représentent le principal intérêt des entreprises. Les clients qui cherchent à externaliser leurs données s'attendent évidemment à être protégés contre les modifications non autorisées. Pour satisfaire cette attente, il est important de réfléchir au contrôle d'accès de ce même service Cloud en implémentant la fonctionnalité de *Role Based Acces Control* ou RBAC. C'est une méthode qui permet de définir un certain nombre d'actions que peut réaliser un utilisateur ou un administrateur au sein du Cloud. En fait, il s'agit de la définition de plusieurs rôles qui auront chacun des permissions et des privilèges différents puis d'associer des collaborateurs, des clients, des partenaires à ces mêmes groupes selon leurs fonctions. C'est donc

une fonctionnalité qui est idéalement présente sur l'ensemble de la chaîne Cloud, des serveurs virtuels, des équipements réseaux jusqu'aux entités virtuelles du stockage.

2.7.1.3 Chiffrement lié aux données

Les défis de la cryptographie dans le Cloud sont complexes, notamment lorsqu'il s'agit de protéger les données hébergées contre des accès non autorisés. Des travaux sont en cours chez les grands offreurs de l'IT, pour la mise en œuvre d'un chiffrement adapté à ces situations et qui fournissent un équilibre satisfaisant entre sécurité, efficacité et fonctionnalité.

Des recherches ont abouti au terme « chiffrement requêtable ». C'est une méthode permettant de manipuler et d'effectuer des recherches sur des données chiffrées sans déchiffrer les données, ainsi que d'en vérifier l'intégrité [1].

2.7.1.4 Données du Cloud accessibles aux autorités d'un pays

Depuis l'affaire Wikileaks, dont les données ont été momentanément stockées sur les services d'Amazon puis retirées très rapidement par ce dernier, la question se pose légitimement. Ce dernier publiait des documents secrets concernant plusieurs pays, surtout sur le domaine politique des Etats-Unis.

Tout pays a le droit légitime d'avoir accès, dans les conditions juridiques qui lui sont propres, aux données qui sont stockées sur son territoire, ou qui transitent par celui-ci. Ainsi, en France, dans le cadre d'une perquisition et conformément au code de procédure pénale, l'hébergeur doit être en mesure d'extraire de son Cloud les éléments recherchés ou l'ensemble des informations concernant un client particulier, sans pour autant avoir à livrer l'ensemble des données des clients hébergés dans le Cloud.

La bonne pratique consiste à assurer contractuellement du ou des pays où seront physiquement installés les éléments d'infrastructures et à connaître, avant de s'engager dans un contrat de service Cloud avec un fournisseur, les juridictions compétentes.

2.7.1.5 Changement de fournisseur

Il serait hasardeux de s'engager dans une solution Cloud sans savoir, à priori, comment on peut la quitter et comment on peut avoir l'assurance que les données, après récupération, seront bien effacées chez le premier prestataire.

Pour faciliter la migration entre prestataire, il existe plusieurs API Cloud émergentes sur le marché, proposées par des consortiums ou des sociétés privées. On privilégiera les API des organismes de standardisation dès lors que ces API auront commencé à être utilisées par un nombre significatif de fournisseurs Cloud. Par exemple, le *Cloud Data Management Interface* (CDMI), provenant du *Storage Networking Industry Association* (SNIA), dont l'objectif est de permettre la portabilité, la conformité, la sécurité des données et l'interopérabilité entre différents fournisseurs Cloud.

De plus, il existe un standard nommé *Open Virtual Machine Format* (OVFM) qui permet une exportation et une importation simplifiée entre différentes plateformes Cloud. Le respect de ce standard assure la compatibilité entre les différentes architectures et permet de changer de fournisseur beaucoup plus librement.

Quant aux données, le processus de réversibilité assure à un client du Cloud qu'elles seront complètement écrasées (ainsi que les sauvegardes) après une période définie contractuellement. Une poursuite judiciaire est possible, selon le pays où le fournisseur s'implante, en cas du non-respect du contrat.

2.7.2 La sécurité logique

A part les données, la partie logique du système doit être aussi sécurisée. Voyons donc quels sont les enjeux et comment sécuriser la partie logique, logicielle, qui compose les infrastructures de services Cloud.

2.7.2.1 La sécurité des serveurs virtuels

Le Cloud s'appuie fortement sur les technologies d'abstraction de services : les clients ne savent pas l'intégralité du système du fournisseur de service. Dans le cadre d'un modèle IaaS, c'est la virtualisation de serveurs qui fournit cette abstraction ; l'élément de base, visible ou non, étant une machine virtuelle (VM) sur un hyperviseur.

La sécurisation des VM consiste à protéger les fichiers des disques virtuels stockés dans le serveur du fournisseur. Pour ce faire, on applique le contrôle d'accès aux serveurs, l'audit et voire le chiffrement. Idéalement, on agit conformément aux recommandations des fournisseurs de l'hyperviseur (configuration des disques virtuels, installation des composants d'intégration dans les VM, etc.).

La seconde famille des bonnes pratiques concerne la notion d'isolation, c'est-à-dire, isolation des flux réseau, isolation des VM par niveau de sécurité et affectation de quotas d'usage des ressources.

2.7.2.2 Sécurisation des accès

Pour maîtriser la sécurité de bout en bout, il faut sécuriser les éléments constituant la plateforme Cloud, mais également l'accès à cette plateforme. Dans le cas d'un accès aux services hébergés sur le Cloud par Internet, le serveur (VM) est nativement vulnérable puisqu'il est directement exposé sur la Toile. Deux solutions de sécurisation peuvent être appliquées :

- Inclure des briques de sécurité de type pare-feu (ouvre les ports applicatifs nécessaires), entre l'infrastructure Cloud et le client, mais aussi éventuellement au sein de l'architecture serveur, entre serveurs front office et serveurs back office.
- Sécuriser chaque serveur virtuel par un pare-feu applicatif installé sur le système d'exploitation. Cela suppose néanmoins une gestion lourde des règles d'accès avec Internet et entre serveurs virtuels.

Si le serveur héberge des services privés de l'entreprise et constitue une extension du SI, il ne doit pas être visible d'Internet, et des solutions de connexions dédiées doivent impérativement être envisagées, telles que :

- La connexion VPN privée : l'accès aux serveurs du Cloud se fait via des liaisons dédiées (fibre, ligne télephonique, etc.) entre le Cloud et l'utilisateur.
- La connexion VPN Internet : l'accès aux serveurs du Cloud se fait via une connexion sécurisée par des mécanismes de chiffrement et d'identification montée entre le Cloud et l'utilisateur, via le transit Internet.

2.7.2.3 Sécurisation de l'administration

L'accessibilité aux interfaces d'administration via une vulnérabilité applicative expose aux risques d'une coupure partielle ou totale du service ou à la perte irrémédiable des données. Par cet accès, l'introduction de virus peut également détériorer les applications, faciliter la corruption des données ou nuire à l'image de marque du service.

Les solutions préventives consistent en la mise en place d'équipement de filtrage (pare-feu, proxy,…) et de solutions antivirus afin de contrôler la légitimité des requêtes entrantes et ainsi garantir l'intégrité des données hébergées. La planification des tests de vulnérabilité et d'intrusion doit ainsi être régulière et fréquente. Cependant, le développement des applications doit être soumis à des audits de codes réguliers, et à l'implémentation de règles et de contrôles des données.

2.8 Avantages

2.8.1 Avantages du Cloud public

Les avantages de cette solution sont nombreux pour l'entreprise:

- Coût de l'énergie : les besoins en énergie ne cesse pas d'augmenter, la facture d'électricité est devenue le principal composant (15 à 20%) du coût total de possession de matériel informatique. L'indicateur d'efficacité énergétique (PUE : Power Usage Effectiveness) tend à baisser davantage dans les grands sites que dans les petits. Les opérateurs de petits datacenter doivent payer l'électricité au tarif local en vigueur, alors que les gros fournisseurs paient moins en implantant leurs datacenters dans des lieux où l'approvisionnement en électricité est moins coûteux et en signant des contrats d'achat en gros.
- Remises quantitatives : les opérateurs de grands datacenter bénéficient souvent d'importantes remises sur les matériels, de l'ordre de 30%, par rapport aux acheteurs habituels [1]. Cela est dû à la standardisation d'un nombre limité d'architectures matérielles et logicielles.
- Flexibilité: les clients peuvent augmenter ou diminuer les ressources à volonté. Ils ne sont pas obligés de surdimensionner leurs systèmes puisqu'ils peuvent ajouter de nouvelles ressources dès que nécessaires. Pour eux, il n'y a donc aucun investissement initial fixe et aucune limite de capacité.
- Nombre de personnels : l'utilisateur ne paie que pour ce qu'il consomme. L'entreprise n'a pas à gérer du personnel pour prendre en charge la gestion des équipements.
- Sécurité et fiabilité : bien que souvent cité comme frein potentiel à l'adoption des Clouds publics, le besoin accru de sécurité et de fiabilité donne lieu à des économies d'échelle. Les principaux fournisseurs commerciaux de Cloud sont souvent mieux armés en la matière. Dotés d'une plus grande expertise qu'un simple service informatique d'entreprise, ils assurent une parfaite sécurité et fiabilité des systèmes du Cloud.

2.8.2 Avantages du Cloud privé

Le Cloud privé permet de garder les données sensibles en interne. Cela a un avantage radicalement différent. Il combine la flexibilité et les avantages en termes de coûts de l'approche du Cloud à une maîtrise intacte de la sécurité et de la fourniture des services. Il offre un chemin de migration judicieux des applications existantes, permettant ainsi de virtualiser et moderniser les applications selon un rythme voulu. Il protège vos investissements dans l'infrastructure, les applications et les informations, tout en permettant aux ressources d'être exploitées plus efficacement et de contribuer à la réactivité de l'entreprise.

2.9 Inconvénients

2.9.1 Inconvénients du Cloud public

Le Cloud public fait perdre de la transparence et par conséquent, le contrôle. Le responsable SI de l'entreprise ignore la plupart du temps l'emplacement des données. Il n'a pas également le moyen de vérifier la méthode de protection de ces derniers. De plus, il ne connait pas les autres fournisseurs non désignés qui participent dans l'ombre au service du Cloud public [7]. Du côté utilisateur, les coûts sont imprévisibles: on paye ce qu'on utilise, alors les factures mensuelles peuvent être différentes chaque mois.

Par ailleurs, les applications existantes de l'entreprise doivent être mises à niveau pour pouvoir s'exécuter dans le Cloud. Ce travail présente trop souvent une charge financière énorme.

2.9.2 Inconvénients du Cloud privé

Le Cloud privé représente moins d'inconvénients que tous les autres. Néanmoins, on peut citer quelques désavantages.

Les matériels utilisés sont appartiennent tous au client. Cela implique que la consommation en énergie et les maintenances sont aussi à leur charge. Dans le cas de changement de la localisation géographique de l'entreprise, le déplacement des ressources est très difficile.

2.10 Conclusion

Le Cloud Computing est une architecture informatique permettant de fournir les ressources comme un service. Ces services se regroupent en trois types principaux (IaaS, SaaS et PaaS) et peuvent être déployés selon quatre modes différents : Cloud public, privé, hybride et communautaire. Ses avantages et inconvénients sont englobés dans le cas du Cloud public et

privé. L'idéal est de pouvoir conjuguer les atouts de ces deux univers, pour pouvoir accéder aux services innovants et à la demande du Cloud public tout en conservant la maîtrise de la gestion comme c'est le cas dans un Cloud privé. L'adoption d'une telle approche hybride apporte une flexibilité exceptionnelle, par la combinaison dynamique des ressources internes et externes. Tout cela va nous servir d'une bonne référence afin d'entamer les prochains chapitres, qui vont nous mener sur l'étude de performance.

CHAPITRE 3

ETUDE DE PERFOMANCE DE CALCUL ET DE TRANSFERT DE DONNEES D'UN SYSTEME INFORMATIQUE

3.1 Introduction

Un système informatique est essentiellement composé par des parties de calcul et des parties de stockage. Sa performance doit suivre la tendance de la demande, ce qui force les administrateurs à ajouter d'autres matériels. Ainsi, la performance du système entier dépend de la qualité de chaque matériel ajouté et de l'architecture en vigueur. Ce chapitre détaillera en premier lieu le processeur et l'architecture des systèmes multiprocesseurs Ensuite, les unités de stockages seront présentées de la même façon, suivie de l'architecture RAID (*Redundant Array of Independent Disks*). On verra à la fin les outils d'évaluation de la performance de calcul et de transfert de données.

3.2 Processeur (Central Processing Unit)

3.2.1 Description

Le processeur est le composant de l'ordinateur qui exécute les « instructions-machine » des programmes informatiques. L'appellation microprocesseur tient du fait que les composants du processeur ne sont plus distincts, mais incorporés dans un même circuit intégré.

Généralement, un processeur est composé d'une unité arithmétique et logique, de plusieurs registres, d'un décodeur d'instruction et d'une interface avec l'extérieur. La synchronisation de toutes les actions de ces éléments est assurée par une horloge. Dans ce cas, la performance d'un processeur dépend principalement de la vitesse d'horloge.

- L'UAL ou Unité arithmétique et logique, prend en charge les calculs arithmétiques élémentaires ;
- Les registres sont des mémoires de petite taille (quelques octets), suffisamment rapides pour que l'UAL puisse manipuler leur contenu à chaque cycle de l'horloge. Un registre d'instruction contient l'instruction en cours de traitement et le registre de données contient les opérandes.
- Le décodeur d'instruction traduit les instructions sous forme de données compréhensibles par l'UAL (opération et opérandes).

- L’interface assure la communication du processeur avec autres éléments extérieurs tels que la mémoire vive, les disques de stockages, les ports E/S, etc.

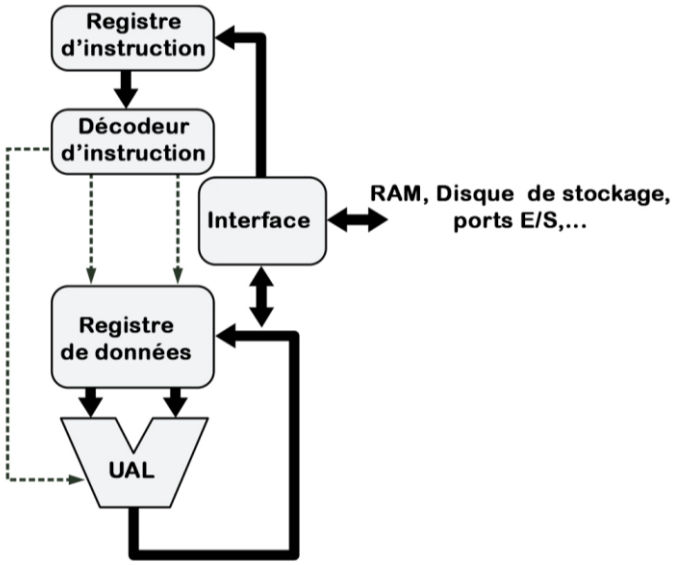

Figure 3.01: *Diagramme simplifié d'un processeur*

En pratique, un programme en cours d'exécution par un ordinateur est représenté par un processus, qui est un ensemble d’instructions à transmettre au processeur. Tous les processus provenant de plusieurs programmes exécutés dans un ordinateur sont donc transmis dans une file d’attente avant d’accéder au processeur. C’est le rôle du système d'exploitation de choisir les processus qui vont être exécutés par les processeurs.

3.2.2 Notion d’opération élémentaire et FLOPS (Floating Point Operations Per Second)

Pour les algorithmes résolvant des problèmes du domaine de l’algèbre linéaire, on utilise souvent le FLOPS comme mesure de dénombrement des opérations élémentaires. Chacune des quatre opérations élémentaires, effectuées en une seconde, à savoir: addition, soustraction, multiplication et division comptent un FLOPS.

Soient les vecteurs $x, y \in \mathbb{R}^n$, $z \in \mathbb{R}^m$ et les matrices $A \in \mathbb{R}^{m\times r}$ et $B \in \mathbb{R}^{r\times n}$ on a alors :

Opération	Nombre de FLOPS
$x + y$	n
$x'x$	$2n\ (n > 1)$
xz'	nm
AB	$2mrn\ (r > 1)$

Tableau 3.01: *Quantification en FLOPS [12]*

3.2.3 Capacité de calcul théorique et puissance maximale

La performance maximale est souvent représentée par la capacité de calcul théorique d'un ordinateur, calculée à partir de la fréquence d'horloge du processeur de la machine, en cycles par seconde, multipliée par le nombre d'opérations par cycle. C'est un point de référence, donné par les constructeurs pour caractériser leurs produits. Pourtant, le rendement réel est toujours inférieur à cette valeur.

On rencontre aussi des microprocesseurs multi-cœurs. C'est un processeur possédant plusieurs cœurs physiques qui travaillent en parallèle. Un cœur physique est en effet un ensemble de circuits capables d'exécuter des programmes de façon autonome. Toutes les fonctionnalités nécessaires à l'exécution d'un programme sont présentes dans ces cœurs : compteur, registres, unités de calcul, etc. Par conséquent, la puissance maximale est proportionnelle au nombre de cœurs du microprocesseur.

$$P_{théorique} = N_{\text{cœurs}} \times f_{horloge} \times N_{opération/cycle} \qquad (3.01)$$

Un microprocesseur ordinaire fait 4 opérations à virgule flottant par cycle. Pour une fréquence de 2,5 GHz, ce dernier a une puissance maximale de 10 GFLOPS (multiplié par le nombre de cœurs s'il s'agit d'un microprocesseur multi-cœur).

3.2.4 Evolution de capacité de calcul d'un système informatique

Face aux demandes accrues, les administrateurs sont obligés à prévoir l'évolution de la capacité de calcul des systèmes informatiques. En effet, cette réalité attire vers une évolution horizontale et verticale.

L'évolutivité horizontale consiste à ajouter des ordinateurs pour faire face à une demande accrue d'un service. La méthode la plus courante est la répartition de charge par utilisation d'une grappe de serveurs (cluster) [19].

L'évolutivité verticale, quant à elle, consiste à utiliser un ordinateur qui offre de nombreuses possibilités d'ajout de pièces, sur lequel il est possible de mettre une grande quantité de mémoire, de nombreux processeurs, plusieurs cartes mères et de nombreux disques durs. Par exemple un ordinateur Sun Enterprise peut contenir jusqu'à 64 processeurs, 16 cartes mères, 64 Go de mémoire et des baies de stockage. L'ensemble tout équipé peut coûter jusqu'à 1 million de dollars [20].

3.2.5 Répartition de charges

Dans le cas d'une grille de calcul, la ressource à exploiter dans tous les nœuds est la capacité de traitement, c'est-à-dire le CPU. Un traitement intensif est morcelé en plusieurs tâches dont chaque nœud prendra part pour le traitement. Néanmoins, il se peut que certaines machines soient sévèrement chargées tandis que d'autres ne sont chargées que faiblement, ce qui serait un signe de faible extensibilité et surtout entrainerait la dégradation de la performance en matière de temps de calcul. Une solution optimisée pour la répartition des tâches peut être découverte en équilibrant les charges entre les nœuds. En pratique, l'équilibrage de charge peut être réalisé en mode statique et dynamique. Chaque nœud est souvent caractérisé par le taux d'utilisation de processeur, le débit de transfert de données, le temps réel d'exécution et le temps d'attente.

3.2.5.1 Ordonnancement statique

Dans l'ordonnancement statique, l'algorithme d'ordonnancement a une connaissance à priori complète de l'ensemble des tâches et de leurs caractéristiques comme les durées d'exécution, les contraintes de précédence et les temps de déclenchement. L'analyse statique conduit à la construction d'une table donnant le début et la fin d'exécution pour l'ensemble des tâches. Entre autres, l'algorithme d'ordonnancement statique opère sur l'ensemble de tâches et produit un schéma d'ordonnancement fixe et invariable. Cette approche n'est valide que pour des systèmes dont on sait qu'ils ne subiront aucune modification ni aucune évolution.

L'algorithme nommé Round-Robin est la plus simple implémentation de l'ordonnancement statique. Il envoie une requête au premier serveur, puis une au second, et ainsi de suite jusqu'au dernier, puis le tour est recommencé au premier serveur, etc. L'ajout d'une pondération fait en sorte que les serveurs de poids fort prennent les premières requêtes et en prennent davantage que ceux de poids faible.

3.2.5.2 Ordonnancement dynamique

Un algorithme d'ordonnancement dynamique a une complète connaissance des tâches actives en cours. De plus, de nouvelles tâches peuvent être activées de façon imprévue si bien que l'ordonnancement varie au cours du temps. Lorsqu'une nouvelle tâche survient, l'ordonnanceur fait une étude de faisabilité afin de déterminer s'il peut garantir l'ordonnancement correct de l'ensemble. Si oui, il établit un schéma d'ordonnancement et insère la tâche dans l'ensemble des tâches garanties.

3.3 Disques de stockages

3.3.1 Mode de lecture/écriture sur un disque de stockage

Un disque de stockage est subdivisé en plusieurs régions, dont chaque région possède une adresse. La lecture/écriture de données peut se faire de façon séquentielle ou aléatoire selon le besoin des utilisateurs et l'organisation de l'unité de stockage en question.

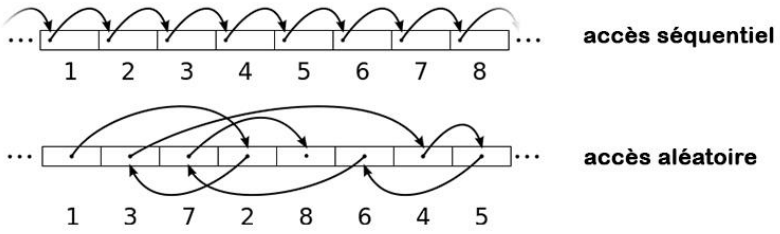

Figure 3.02: *Mode de lecture/écriture sur un disque de stockage*

3.3.2 Notion d'IOPS (Input/Output Operations Per Second)

IOPS est une unité de mesure de performance utilisée pour référencer les appareils de stockage informatique tels que les disques durs, les disques électroniques, et les réseaux de stockage. Un test standard utilise un bloc de données de 4Ko, avec un mode de lecture/écriture aléatoire. Le résultat peut alors être exprimé en fonction du débit de transfert du disque et la taille du bloc de données.

$$Nombre_{\boldsymbol{IOPS}} = \frac{Débit_{E/S}}{Taille_{bloc}} \qquad (3.02)$$

3.3.3 Disque dur (Hard Disk Drive)

3.3.3.1 Description

Un disque dur est un dispositif de stockage de données utilisé pour stocker des informations numériques en utilisant des disques en rotation rapide (plateaux) revêtue d'un matériau magnétique. Il peut avoir un ou plusieurs plateaux; chaque plateau a deux côtés, dont chacun est appelé une surface. Les plateaux sont tous liés ensemble à un moteur qui tourne à une vitesse constante. Les données sont codées sur des pistes qui sont subdivisées en plusieurs secteurs et délimitées par des cercles concentriques. Or, une seule surface contient des milliers de pistes

serrées ensemble. La lecture et l'écriture sur la surface sont effectuées par une tête, positionnée sur un mécanisme de déplacement.

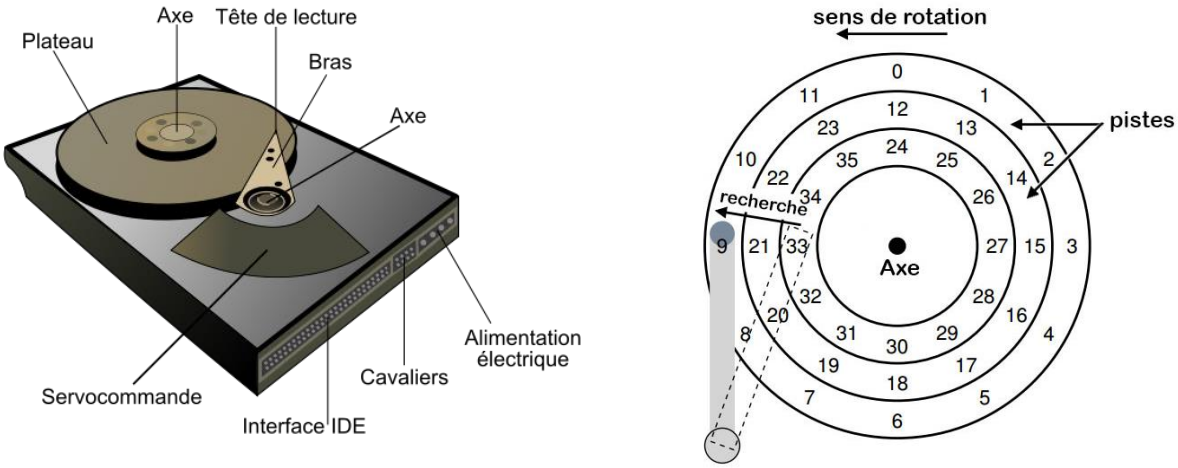

Figure 3.03: *Structure d'un disque dur*

3.3.3.2 Latence de rotation ($\boldsymbol{L_{rotation}}$)

Pour une position de la tête sur une piste, la latence de rotation désigne le délai d'attente d'une donnée voulue. Un disque qui tourne à 10000 RPM signifie qu'une seule rotation dure environ 6 millisecondes (6 ms).

3.3.3.3 Temps de recherche d'une piste ($\boldsymbol{T_{recherche}}$)

La recherche d'une piste se divise en quatre phases : phase d'accélération, phase de déplacement constante, phase de décélération et phase de positionnement. La totalité de la durée de ces quatre phase est représenté par $T_{recherche}$.

3.3.3.4 Temps d'accès

Le temps d'accès désigne la durée nécessaire pour pointer sur une adresse avant la lecture ou l'écriture. Ce temps d'accès est requis avant de procéder à la lecture ou à l'écriture.

$$T_{accès} = T_{recherche} + L_{rotation} \qquad (3.03)$$

3.3.3.5 Temps de transfert ($\boldsymbol{T_{transfert}}$)

Un disque dur est aussi doté d'un mémoire cache de l'ordre de méga octets. Ce cache joue aussi le rôle d'une file d'attentes des données à écrire sur le disque. Le temps de transfert est obtenu à

partir de la dimension du bloc de données (Dim_{bloc}) à transmettre et du débit de transfert ($\text{Débit}_{\text{cache}}$) vers la mémoire cache, donné par le constructeur.

$$T_{transfert} = \frac{Dim_{bloc}}{Débit_{cache}} \tag{3.04}$$

3.3.3.6 Temps d'opération E/S ($T_{E/S}$)

Une opération E/S est définie comme un échange de donnée élémentaire entre un système électronique et un disque dur. Ainsi, le temps d'opération E/S représente la durée totale de l'opération d'écriture ou de lecture de donnée sur un disque dur.

$$T_{E/S} = L_{rotation} + T_{recherche} + T_{transfert} \tag{3.05}$$

3.3.4 Disque électronique (Solid-State Drive)

3.3.4.1 Description

Un SSD est un matériel informatique permettant le stockage de données sur des blocs de mémoires flash. Il est composé d'une interface vers le système hôte, un contrôleur SSD, une mémoire tampon et des blocs de mémoires flash. De ce fait, le contrôleur SSD assure la synergie de tous les composants du disque. De son côté, la mémoire tampon joue le rôle d'une file d'attente durant l'échange de données entre le système hôte et le contrôleur SSD. Chaque bloc de mémoire flash est composé par des millions de transistors sous forme NAND (*Negative-AND*) en raison du besoin de rapidité en lecture/écriture.

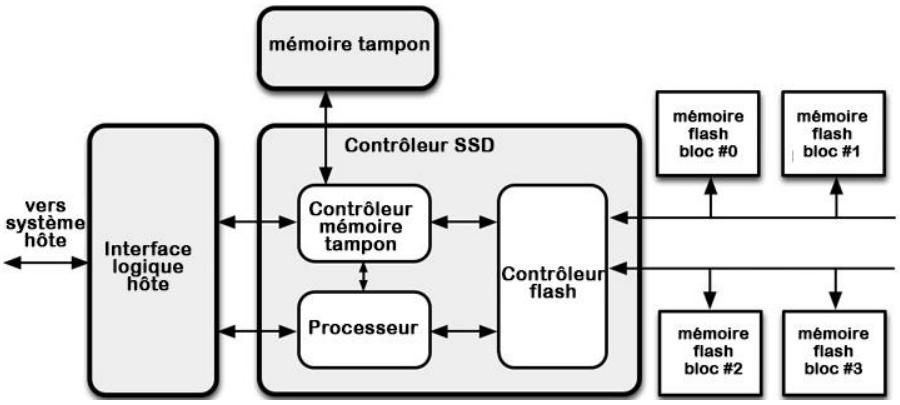

Figure 3.04: *Architecture d'un SSD*

3.3.4.2 Techniques utilisées

a). FGMOS (Floating-Gate MOSFET)

Un FGMOS est un transistor à effet de champ, dont la structure est semblable à un transistor MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*) classique. Il est doté d'une grille flottante isolée, capable de stocker une charge négative. Chaque niveau de charge définit une courbe caractéristique du FGMOS. L'application d'une tension positive importante entre la grille et la source réduit le niveau de charge de la grille flottante à 0 tandis que l'application d'une tension négative importante rend le niveau de charge de la grille flottante au niveau maximal.

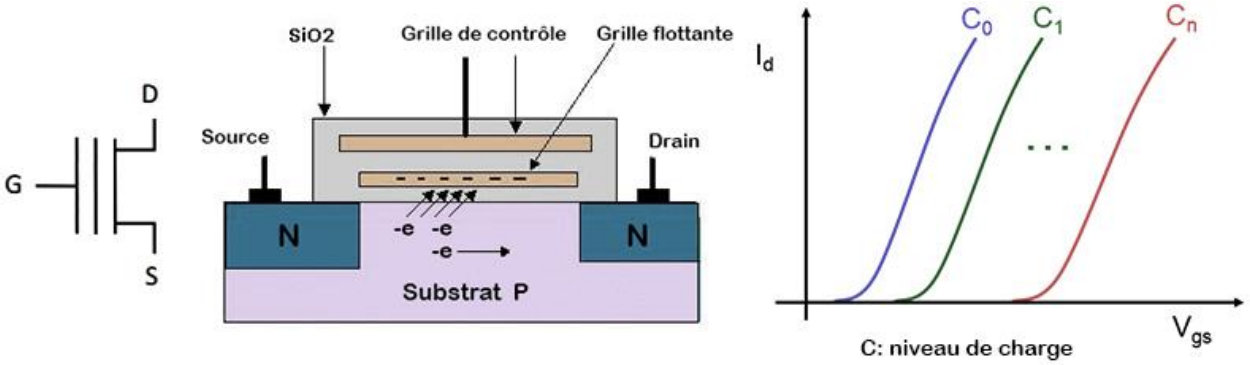

Figure 3.05: *Symbole, structure et courbe caractéristique d'un FGMOS*

b). Mémoire flash SLC NAND (Single Level Cell Negative-AND)

Le SLC désigne la façon dont on fait correspondre une cellule élémentaire à un bit (deux niveaux de charge). Ainsi, une mémoire flash de n bits nécessite n FGMOS pour le stockage des données et 2 MOSFET pour le sélecteur de bit et de masse.

La lecture/écriture sur une cellule se fait à l'aide de 5 niveaux de tension : V_{off}, V_{on}, V_{in}, $-V_{high}$ et V_{high}. Entre autres, l'application des tensions V_{off} et V_{on} sur la grille de contrôle entraine respectivement le blocage et la conduction du FGMOS quel que soit la charge contenue dans la grille flottante. Le niveau de tension V_{in} est utile pour la lecture de l'information dans la cellule. L'écriture se fait à son tour à l'aide de $-V_{high}$ et V_{high}.

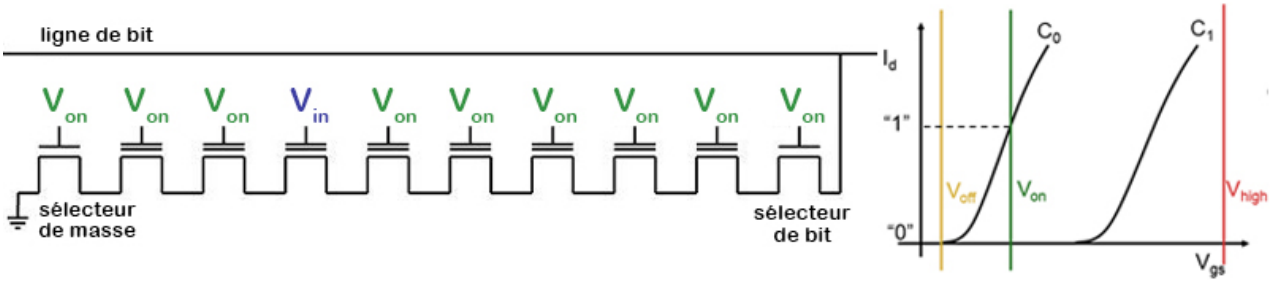

Figure 3.06: *Lecture du 3ème bit dans une mémoire flash SLC NAND[39]*

La phase d'écriture se divise en deux. Tout d'abord, la réinitialisation de tous les bits en « 1 », puis l'écriture de chaque bit « 0 ».

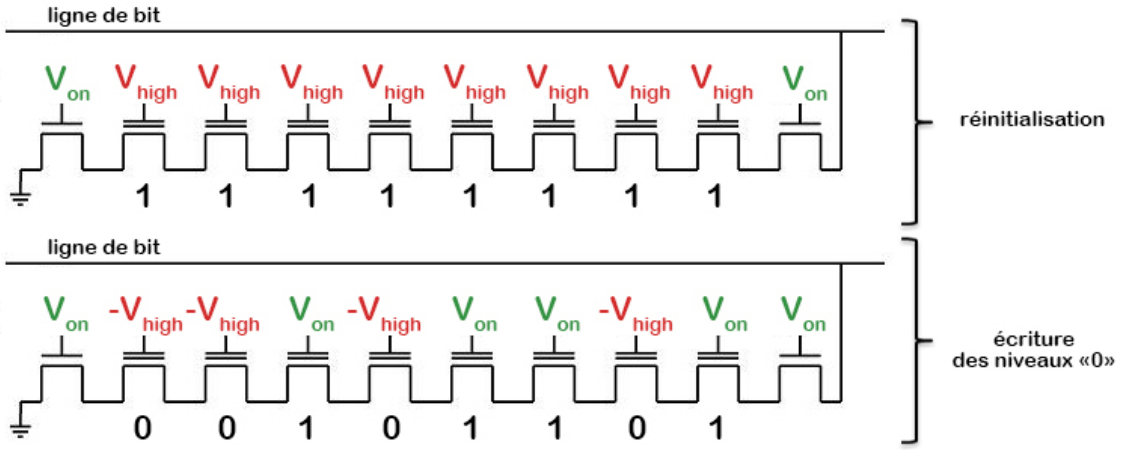

Figure 3.07: *Ecriture d'une information sur une mémoire flash SLC NAND[39]*

Un bloc de mémoire est obtenu à partir d'un assemblage de plusieurs mémoires flash de quelques bits. L'adressage d'une cellule élémentaire se fait à l'aide d'un décodeur ligne et décodeur colonne.

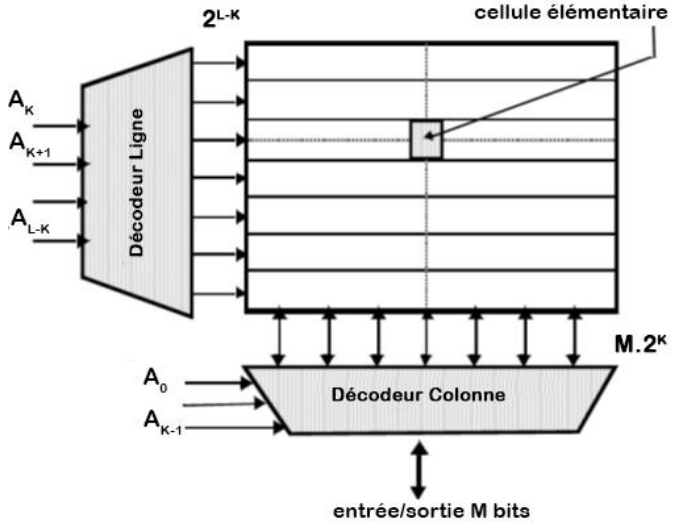

Figure 3.08: *Bloc de mémoire flash*

c). Mémoire flash MLC NAND (Multi Level Cell Negative-AND)

La MLC NAND fait correspondre plusieurs bits sur chaque cellule élémentaire. La lecture/écriture nécessite beaucoup plus de niveau de tension et de nombre de tests. Cela réduit non seulement la densité de transistor utilisé, mais aussi la performance et la durée de vie de la mémoire. Une mémoire flash SLC NAND a une durée de vie environ 100.000 cycles d'écritures tandis qu'une mémoire MLC n'en a que 1000 à 10.000 cycles d'écritures.

3.3.4.3 Tableau comparatif

Caractéristique	SSD	HDD
Temps d'accès aléatoire	Environ 1ms	2,9 à 20ms
Vitesse de lecture/écriture maximale	96 Mo/s à 4,8 Go/s	40 à 260 Mo/s
IOPS	2400 à 1.200.000	30 à 560

Tableau 3.02: *Comparaison entre la performance d'un HDD et SSD[17]*

3.3.5 RAID (Redundant Array of Independent Disks)

3.3.5.1 Description

L'acronyme RAID, défini en 1987 par l'Université de Berkeley, désigne les techniques permettant de répartir des données sur plusieurs disques durs afin d'améliorer soit les performances et la tolérance aux pannes de l'ensemble du système. De ce fait, une partie de la capacité physique est utilisée pour stocker de l'information redondante concernant les données utilisateurs. Cette information redondante permet la régénération des données d'utilisateurs perdues lorsqu'une unité ou un chemin de données à l'intérieur du système est défaillant.

En 1998, il existait 5 niveaux de RAID, numérotés de 1 à 5, correspondant aux différents niveaux de fiabilité et de performance d'un système. Depuis, d'autres types de RAID sont apparus. Ils sont soit l'évolution de RAID déjà existants, soit une combinaison de plusieurs niveaux de RAID [16]. Les niveaux 1, 3, 4 et 5 sont classés comme type standard (le niveau 2 est aujourd'hui obsolète).

3.3.5.2 Les différentes types de systèmes RAID

Un système RAID peut être réalisé de trois façons différentes :

- RAID logiciel : réalisation sous un ordinateur doté de plusieurs disques de stockage ;
- RAID matériel : réalisation dans un matériel dédié ;
- RAID pseudo matériel : réalisation dans une carte mère spécifique.

3.3.5.3 Critères d'évaluation d'un RAID

Le RAID affecte plusieurs paramètres dans l'ensemble du système de stockage. On rencontre trois principaux critères pour l'évaluation de chaque conception [15] :

- La capacité utile parmi N disques utilisés ;
- La fiabilité, représentée par le nombre de disques en panne toléré ;
- La performance en lecture/écriture.

Par ailleurs, les paramètres suivants sont considérés comme critères secondaires : consommation en énergie, le taux d'utilisation de CPU, le coût de la réalisation.

3.3.5.4 RAID-0

Le RAID-0 n'est pas réellement un RAID dans la mesure où il ne répond pas exactement à la définition d'un RAID. Dans ce système, il n'y a aucune répétition d'informations, donc aucune redondance, ce qui implique qu'il n'y a aucune sécurité face aux pannes. Ce mode consiste juste à unifier plusieurs disques et à répartit les données sur plusieurs disques.

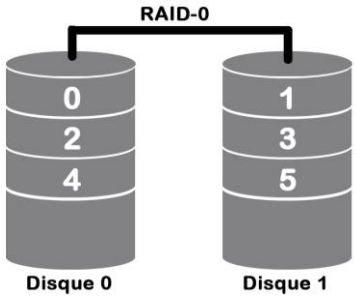

Figure 3.09: *RAID-0*

3.3.5.5 RAID-1 : mirroring

Tout fonctionne par paires de disques durs. Lorsque l'on écrit une donnée, on l'écrit sur deux disques de sorte que l'on obtient deux disques identiques au niveau des données. Si jamais un des deux disques durs tombe en panne, l'autre contient toutes les informations, d'où le très bon niveau de sécurité.

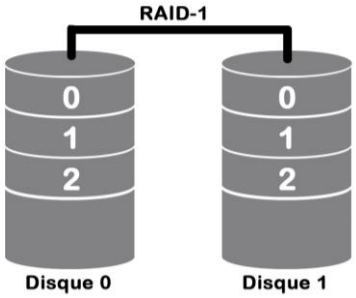

Figure 3.10: *RAID-1*

3.3.5.6 RAID-3 et RAID-4

La technique du RAID-3 et RAID-4 utilise le principe du RAID 0 auquel on associe un disque supplémentaire afin de stocker des bits de parité. L'opérateur utilisé pour ce dernier est le XOR. Si un disque tombe en panne, il est possible, à partir du disque de parité, de reconstituer l'information présente sur le disque défectueux.

La différence réside au niveau de la segmentation des données. Dans un RAID 3, la segmentation des données s'effectue au niveau des octets, tandis que dans un RAID 4, on effectue une segmentation par bloc. De plus, le RAID-3 nécessite une synchronisation des disques. La lecture et l'écriture s'effectuent simultanément sur tous les disques. Pour le RAID-4, les données sont décomposées en segments de taille variable, d'un ou plusieurs secteurs. En effet, cela permet d'éviter la synchronisation et d'augmenter la performance.

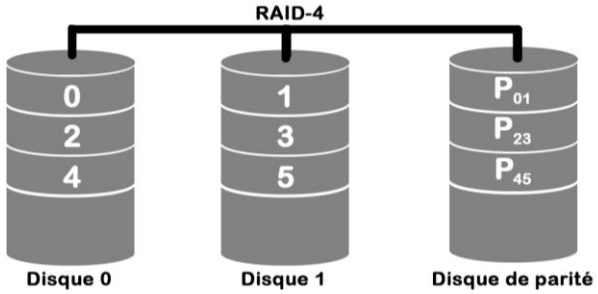

Figure 3.11: *RAID-4*

3.3.5.7 RAID-5

Le RAID 5 est le plus utilisé des niveaux de RAID [16]. Si on utilise N disques, chaque disque sera segmenté de plusieurs partie de N segments ($N \geq 2$) : ($N - 1$) segments de données de un segment de parité. Les segments de données d'un disque sont ensuite répartisses dans les autres disques. En cas de panne d'un disque, il est possible de le changer, les données étant alors reconstruites à partir de celles des autres disques.

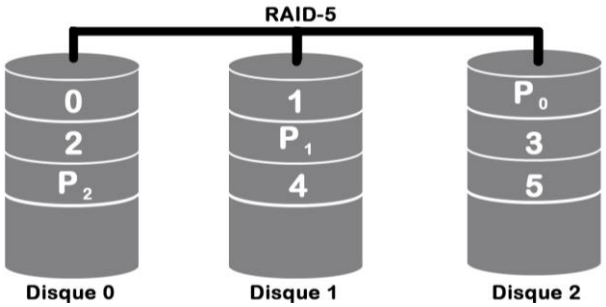

Figure 3.12: *RAID-5*

3.3.5.8 Capacité, fiabilité et performance des systèmes RAID standards

Le tableau de récapitulation ci-dessous correspond à des systèmes RAID composés de N disques identiques :

- Capacité (C) ;
- Vitesse de lecture/écriture séquentielle (V_{LS}, V_{ES}) ;
- Vitesse de lecture/écriture aléatoire (V_{LA}, V_{EA}).

Type		**RAID-0**	**RAID-1**	**RAID-4**	**RAID-5**
Capacité		N × C	$\frac{N}{2} \times C$	(N – 1)×C	(N – 1)×C
Nombre de disque défaillant toléré		$\frac{N}{2}$	1	1	1
Performance	**Lecture séquentiel**	N × V_{LS}	$\frac{N}{2} \times V_{LS}$	(N – 1)× V_{LS}	(N – 1)× V_{LS}
	Ecriture séquentiel	N × V_{ES}	$\frac{N}{2} \times V_{ES}$	(N – 1) × V_{ES}	(N – 1) × V_{ES}
	Lecture aléatoire	N × V_{LA}	N × V_{LA}	(N – 1) × V_{LA}	(N – 1) × V_{LA}
	Ecriture aléatoire	N × V_{EA}	$\frac{N}{2} \times V_{EA}$	$\frac{1}{2} \times V_{EA}$	$\frac{N}{4} \times V_{EA}$

Tableau 3.03: *Capacité, fiabilité et performance des systèmes RAID standard [15]*

3.4 Choix des outils d'évaluation de performance

La performance en calcul et en transmission de données peut être évaluée par plusieurs outils différents. La base des tests repose toujours sur la rapidité et la quantité d'opérations traitées.

Le premier problème concerne la fiabilité du test, c'est-à-dire, la capacité de l'outil à donner la valeur précise de la performance d'un système informatique.

Le second problème réside sur l'unité des résultats. On rencontre souvent des résultats sous forme de nombre de points. D'autres outils de test retournent le temps écoulé durant l'évaluation comme résultat. Tous ces résultats peuvent indiquer la performance, mais avec des unités propres pour chaque outil.

On a choisi LINPACK et HD Tune comme outils d'évaluation de performance en raison de l'unité de résultat qu'ils fournissent. A part cela, ils offrent aussi la possibilité de changement de paramètres pour assurer la fiabilité des tests.

3.4.1 LINPACK

3.4.1.1 Description

LINPACK est un outil servant à tester la performance de calcul des ordinateurs. C'est une application multiplateforme (fonctionnant sur Windows, Linux, Mac OS…), basée sur la résolution de systèmes d'équations linéaires. Elle donne directement la capacité de calcul du système en FLOPS.

3.4.1.2 Historique

En 1979, LINPACK a été conçu pour aider les utilisateurs à évaluer le temps requis par leurs systèmes pour résoudre un problème. A ce moment, la taille de la matrice était limitée à 100, en raison de la capacité mémoire.

Depuis 1993, LINPACK était utilisé pour classer les plus puissants superordinateurs du monde dans le TOP500 [10]. C'est un classement organisé par l'Université du Tennessee (USA) et l'Université de Mannheim (Allemagne).

La performance des superordinateurs n'a cessé de croître, doublée environ tous les 14 mois. En novembre 2014, le Tianhe-2 figure sur le premier rang du classement, avec une puissance de 54,9024 PFLOPS. C'est plus de 419102 fois plus rapide que le système le plus rapide en juin 1993 : la machine CM-5, avec une puissance de 59,7 GFLOPS [11].

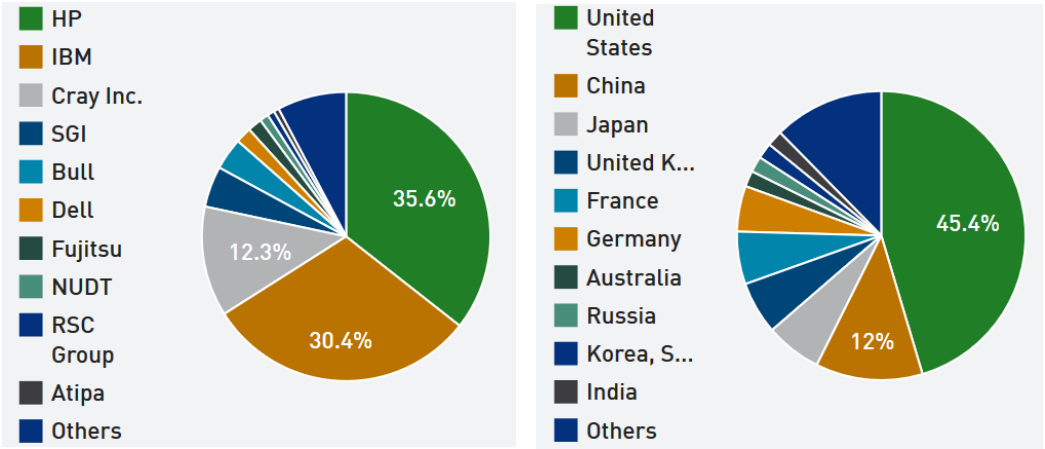

Figure 3.13: *Répartition des 500 supercalculateurs mondiales*

3.4.2 Principe

Pour fournir une bonne correction sur la performance maximale fournie par le fabricant, LINPACK évalue la capacité d'un ordinateur à résoudre un système de n équations à n inconnues. Il mesure le temps mis par l'ordinateur en question pour la résolution du système d'équations. La performance est ensuite calculée en divisant le nombre d'opérations par le temps dépensé par la résolution, donc en FLOPS.

Un système de n équations à n inconnues $(x_1, x_2, \dots, x_n)$ peut être représenté comme suit:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_n + = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2p}x_n + = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{np}x_n + = b_n \end{cases} \tag{3.06}$$

$$\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & \dots & a_{2p} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ \dots \\ b_n \end{pmatrix} \Leftrightarrow A_{nn}X_n = B_n \tag{3.07}$$

3.4.3 Nombre d'opérations élémentaires

Un système de n équation à n inconnues peut être résolu selon plusieurs méthodes différentes. On rencontre : la méthode de Cramer, l'inversion matricielle et le pivot de Gauss.

La précision du test de performance en calcul se base sur la connaissance du nombre d'opérations élémentaires nécessaires pour la résolution du système d'équation.

LINPACK utilise la résolution par pivot de Gauss où le nombre d'opération est égale à N_{op}[12].

$$N_{op} = \frac{2}{3}\, n^3 + \frac{3}{2}\, n^2 \tag{3.08}$$

Pour une valeur de n élevée, n^2 deviendra négligeable devant n^3. On peut donc simplifier l'équation ci-dessus.

$$N_{op} \simeq \frac{2}{3}\, n^3 \tag{3.09}$$

La puissance réelle d'un système informatique testé par LINPACK est alors :

$$P_{réelle} = \frac{N_{op}}{t} \tag{3.10}$$

3.5 Cas pratique

L'utilisation de LINPACK nécessite la définition de quatre paramètres: le nombre d'équations à résoudre, le nombre de tests à effectuer et le mode d'alignement en mémoire.

3.5.1 Le nombre d'équations à résoudre (n)

Le nombre d'équations à résoudre définit la dimension (n) du système d'équations. Ce nombre a beaucoup d'impact sur le temps d'exécution de l'évaluation : le nombre d'opérations élémentaires nécessaires à la résolution du système d'équations est de l'ordre de n^3. Par contre, la durée d'évaluation trop brève (inférieur à une seconde) augmente la marge d'erreur.

3.5.2 Le leading dimension (l)

La matrice représentative du système d'équations en a « l » ligne et « n » colonne. Pour cela, la capacité mémoire utile pour le stockage est donc représentée comme suit :

$$C_{mém} = N_{enc} \times n \times l \tag{3.11}$$

N_{enc} représente le nombre de bit servant à encoder une chiffre à virgule flottante (IEEE 754).

$N_{enc} = 32bits$ pour une simple précision ;

$N_{enc} = 64bits$ pour une double précision.

3.5.3 Nombre de tests

Nombre de test à effectuer permet de lancer le test plusieurs fois. Les résultats obtenus sont analysés et la puissance maximale, moyenne et minimale sont indiquées à la fin de tous les tests.

3.5.4 Mode d'alignement en mémoire

Le mode d'alignement en mémoire représente la façon dont la machine organise et accède dans la mémoire vive. Quand un ordinateur moderne lit ou écrit à une adresse mémoire, il le fait par bloc de 32 bits, 64 bits ou plus (4 octets, 8 octets, …). Pour un ordinateur 32 bits (4 octets), les données doivent être lues sur des adresses multiples de 4. Ainsi, une donnée d'un octet sur l'adresse mémoire 14 nécessite la lecture de des données sur l'emplacement 12 à 15 et 16 à 19. La notation en n Ko représente un bloc d'alignement de n octets.

3.5.5 HD Tune

3.5.5.1 Description

HD Tune est un utilitaire de stockage, propriétaire et fonctionne seulement sur Windows. Son fonctionnement principal se focalise sur la mesurer la performance en lecture et écriture d'un disque de stockage et l'identification des éventuelles les erreurs. La version actuelle (v5.5) en a les fonctions suivantes [12]:

- Benchmark : mesure la performance en lecture/écriture d'un disque de stockage ;
- File Benchmark: mesure la performance en lecture/écriture d'une partition ;
- Random Access: mesure du temps d'accès (lecture/écriture) d'une partie aléatoire d'un disque de stockage;
- Extra Tests: mesure rapide de la performance (lecture/écriture) d'un disque de stokage ;
- Info : indication des informations détaillées d'un disque de stockage (capacité, numéro de série…) ;
- Health : indication de santé d'un disque de stockage selon le système de surveillance SMART (*Self-Monitoring, Analysis, and Reporting Technology*) ;
- Error Scan : scan des secteurs défectueux sur un disque de stockage ;
- Erase: suppression sécurisée des données dans un disque de stockage ;
- Disk Monitor: monitoring d'un disque de stockage ;
- Folder Usage: indication de la répartition dossiers dans un disque de stockage ;
- AAM (*Acoustic Mismatch Model*): réglage entre bruits et la performance d'un disque dur;
- Temperature display : indication de la température d'un disque de stockage.

Parmi toutes les fonctionnalités d'HD Tune, nous nous intéressons seulement sur la méthode d'évaluation de santé et de performance des unités de stockage.

3.5.5.2 Evaluation de santé d'une unité de stockage

Les disques durs peuvent souffrir de deux types de défaillances : les défaillances prévisibles, les défaillances imprévisibles. Les défaillances prévisibles surviennent suite à la dégradation lente de certains composants, en particulier à cause de l'usure et du vieillissement des composants du disque. Les défaillances imprévisibles, quant à elle peuvent survenir soudainement, comme un composant électrique qui grille.

Un système de surveillance SMART peut détecter le premier type de défaillance, tout comme la jauge de température du tableau de bord d'une voiture peut prévenir le conducteur avant que de graves dégâts n'apparaissent. Il fait périodiquement un diagnostic selon plusieurs indicateurs de fiabilité dans le but d'anticiper les erreurs sur le disque dur. Selon la statistique, 30% des pannes peuvent être prévues par le système SMART dont les 60% sont dues aux défaillances du système mécanique [14].

Les caractéristiques suivantes figurent parmi les importantes informations du résultat d'un diagnostic de disque dur :

- La durée de fonctionnement ;
- Le nombre de mis en mâche et mis en arrêt ;
- Le nombre d'écritures pour les SSD ;
- La température pour les HDD.

3.5.5.3 Evaluation de performance d'un disque dur

L'évaluation de performance d'un disque dur entier est assurée par la fonction « benchmark ». Elle évalue la capacité en lecture et écriture selon différents blocs de données prédéfinies, d'une manière explicite, elle fait une lecture ou une écriture par bloc de données qui varient de 512 octets à 8Mo. Par conséquent, le taux de transfert est proportionnel à la taille du bloc de données à lire et à écrire.

a). Taux de transfert

Le taux de transfert de données en lecture est donné après un balayage total du disque dur tandis que le taux de transfert de données en écriture est donné après un balayage partiel pour éviter la perte des données. Le balayage se fait à son tour, à un intervalle constant. La durée de l'évaluation dépend de la longueur de cet intervalle. Pour le HDD, la lecture et l'écriture commencent par les pistes extérieures et se déplacent vers les pistes intérieures. Cela peut être vu dans le graphique de la vitesse de transfert où la vitesse est la plus élevée au début de l'essai et diminue vers la fin de l'essai.

Le taux de transfert minimal et maximal est aussi indiqué durant le traçage de la courbe de performance. La moyenne est aussi indiquée à fin de l'évaluation.

b). Temps d'accès

Le temps d'accès à une donnée sur une unité de stockage est mesuré en milliseconde (ms). C'est la durée nécessaire pour pointer sur une adresse avant la lecture ou l'écriture. Chaque durée d'accès est représentée par un pointillé jaune pendant l'évaluation de performance et le temps d'accès moyen est donné à la fin de la représentation.

c). Capacité de transfert des interfaces

La capacité de transfert des interfaces des unités de stockages (IDE, SATA, SCSI,…) représente le taux de transfert maximal entre le système d'exploitation et le périphérique en question. Elle est aussi donnée à la fin de l'évaluation de performance.

d). Taux d'utilisation de CPU

La lecture et l'écriture sur une unité de stockage nécessitent une part de travail du CPU. Le résultat est donné en pourcent, selon la capacité de calcul maximale du CPU.

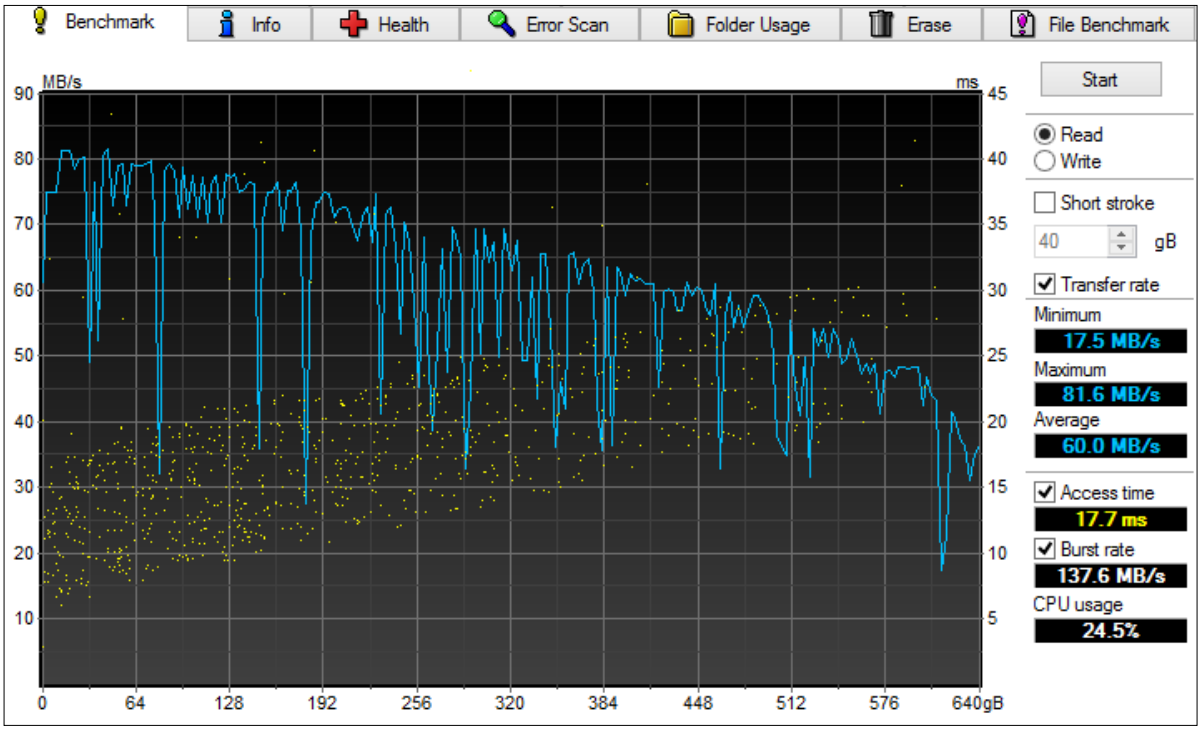

Figure 3.14: *Aperçu de l'évaluation de performance d'un disque dur*

3.5.5.4 Evaluation de performance d'une partition

Le test de taux de transfert d'une partition mesure trois paramètres différents à la fois pour la lecture et l'écriture.

- vitesse séquentielle : mesurée et représentée sur le graphique. Idéalement, le transfert à grande vitesse doit être droit et lisse. Lorsque le test est terminé, la vitesse de transfert moyenne est affichée ;
- 4 Ko aléatoire simple: c'est une évaluation de performance des opérations d'E/S de 4096 blocs d'octets. C'est une opération d'E/S le plus commun sur un système typique, surtout la vitesse d'écriture de 4 Ko est une indication importante de la performance générale du système. A la fin du test, le résultat est donné en IOPS ;
- 4 Ko aléatoire multiple: ce test est similaire à la « 4 Ko aléatoire simple », sauf que des blocs multiples 4Ko sont attribués simultanément au dispositif. Le nombre d'opérations peut être spécifié et peut avoir une valeur comprise entre 2 et 64.

Une plus grande taille de fichier de test donnera des résultats plus précis. De plus, le format de donnée (data pattern) utilisé pendant l'écriture peut être spécifié. Les choix sont zéro, aléatoire et mixte qui est une combinaison de zéros et des données aléatoires. Cependant, certains SSD utilisent une technique de compression qui améliore les performances lorsque des données compressibles sont utilisées. Pour ces dispositifs, les résultats seront plus élevés lors de l'écriture des zéros et le plus faible lors de l'écriture des données aléatoires.

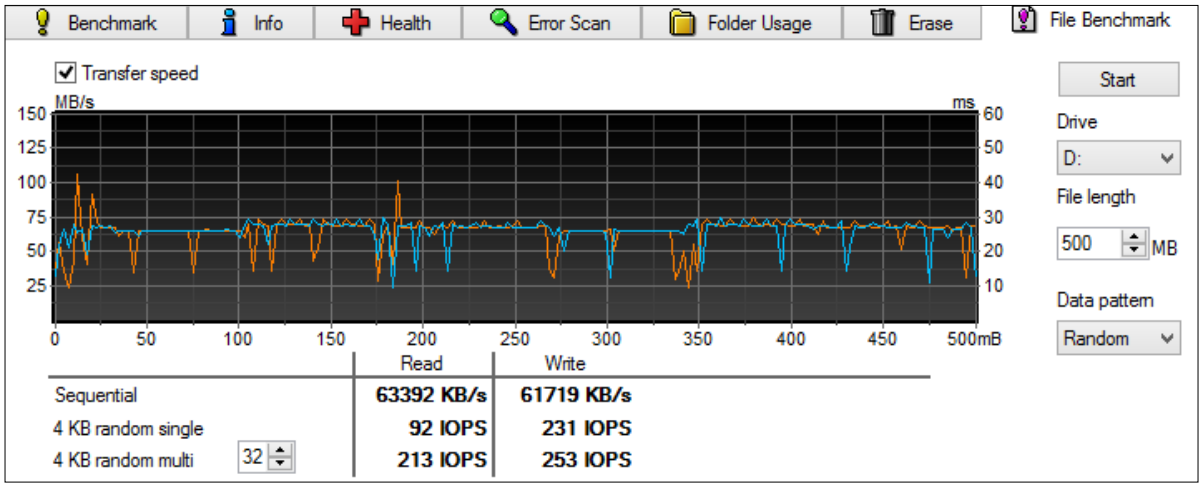

Figure 3.15: *Aperçu de l'évaluation de performance d'une partition*

3.6 Conclusion

En résumé, un système monoprocesseur peut évoluer de façon verticale ou horizontale : ajout d'unités de calcul (CPU) et de serveurs. Quant aux unités de stockages, elles peuvent être rassemblées selon plusieurs niveaux RAID pour améliorer la performance en lecture/écriture et la fiabilité des données. Face aux performances théoriques, LINPACK et HD Tune permettent d'évaluer respectivement la performance de calcul et la performance de transfert de données d'un système informatique.

CHAPITRE 4

DEPLOIEMENT ET EVALUATION D'UN CLOUD PRIVE SOUS OPENSTACK

4.1 Introduction

Un Cloud privé est une architecture déployée dans des serveurs physiques. Sa performance dépend de la performance de ces derniers et de l'outil utilisé pour son déploiement. Ce chapitre représentera OpenStack et la façon dont on a déployé un Cloud privé sur deux serveurs. On présentera à la fin la mesure de performance de calcul et de stockage de chaque entité (serveurs physiques et machines virtuelles), ainsi que l'interprétation sur le rendement de virtualisation.

4.2 Choix et présentation d'OpenStack

OpenStack est un ensemble de logiciels libres qui permet la construction de Cloud privé et public. En juillet 2010, son développement a été initié par la NASA, l'agence gouvernementale qui prend en charge la majeure partie du programme spatial civil des Etats-Unis et Rackspace Cloud, un fournisseur Cloud. Ces deux entreprises ont été rejointes par : Canonical, Red Hat, Cisco, Dell, HP, IBM, Yahoo, Oracle, Orange, Cloudwatt, VMware, Intel, etc.

Dans le domaine du logiciel libre, deux autres solutions peuvent remplacer OpenStack : Eucalyptus et Open Nubla. Face à eux, OpenStack en a les points forts suivants [24] :

- Stabilité : projets testés et corrigés périodiquement;
- Excellente documentation : facilité d'accès et regroupement sous forme wiki ;
- Très bonne sécurité : un projet dédié à la sécurité ;
- Communauté accueillante : toujours prête à répondre aux questions.

Sur le fond, il possède une architecture modulaire composée de plusieurs projets qui gèrent le contrôle des différentes ressources des machines virtuelles telles que la puissance de calcul, le stockage ou encore les connexions au réseau. En effet, cette subdivision en plusieurs modules facilite la mise en œuvre d'un Cloud et l'extension de l'infrastructure. La version « icehouse », sortie en avril 2014 est composée par 8 projets principaux : Horizon, Keystone, Nova, Glance, Cinder, Swift, Neutron et Ceilometer.

La figure suivante montre Architecture détaillée d'OpenStack avec ces projets cités précedemment.

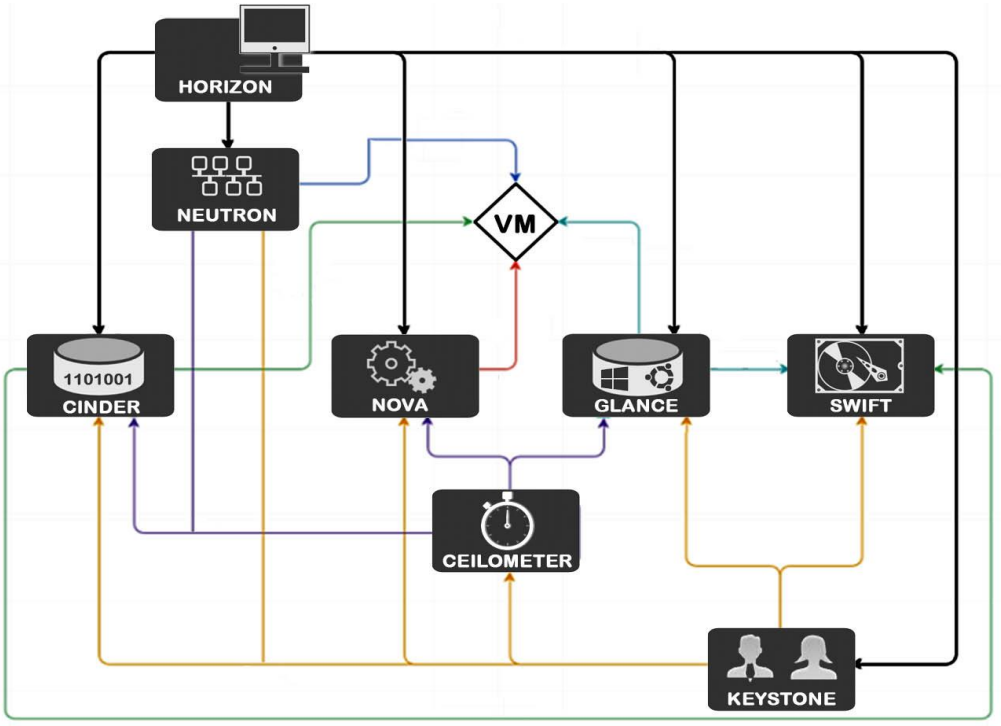

Figure 4.01: *Architecture détaillée d'OpenStack[21]*

4.2.2 Projet Horizon

Horizon offre une interface web de la plupart des composants d'OpenStack. L'accès se fait à partir d'un navigateur web. Il facilite l'administration du Cloud en affichant les états des composants et en exécutant plusieurs commandes en un seul clic. Ainsi, l'administrateur a un accès à des différents menus:

- *Vue d'ensemble*: récapitulatif de l'usage des serveurs par projet, utilisation actuelle en nombre de CPU virtuels, RAM (Memoire RAM), et Disques de stockage puis compteur en CPU et espace disque (Go) par heures ;
- *Instances*: gestion des instances des machines virtuelles plus quelques infos globales comme le projet auquel elles appartiennent, le serveur hôte, l'adresse IP, la taille, le statut et les actions en cours ;
- *Type d'instance*: gestion des types d'instances suivant leurs spécifications en nombre de CPU, mémoire vive (Memoire RAM), espace disque ;
- *Images* : gestion des images des disques virtuels stockées par le service Glance ;

- *Projets*: gestion des projets existants et leurs statuts ;
- *Utilisateur*: gestion des utilisateurs ;
- *Réseau* : gestion de l'interconnectivité des entités machines virtuelles à l'aide d'un réseau virtuel.

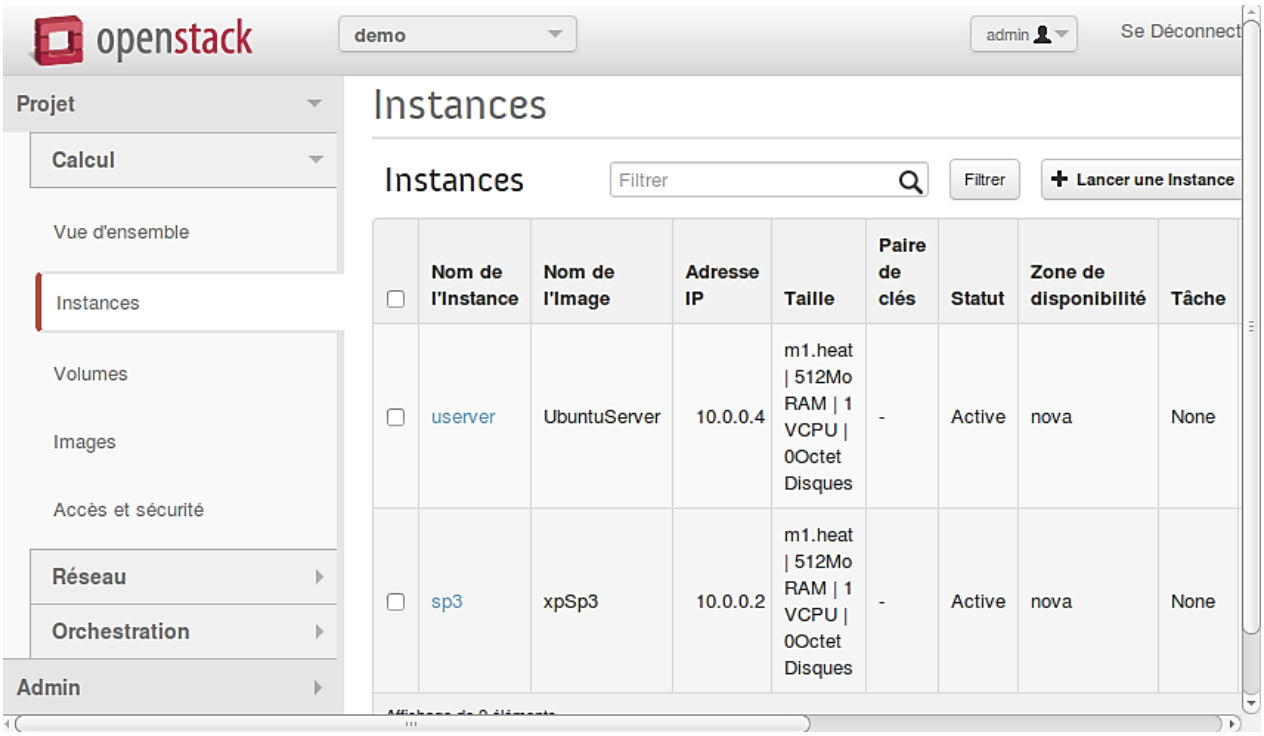

Figure 4.02: *Aperçu du projet Horizon*

4.2.3 Projet Keystone

Keystone gère l'autorisation et l'authentification de chaque module d'OpenStack. A part cela, il possède aussi un catalogue qui contient tous les services que chaque composant d'OpenStack peut offrir.

La gestion des utilisateurs s'articule autour de 3 objets:

- l'objet *User* qui est une représentation numérique d'une personne, un système ou un service qui utilise les ressources d'OpenStack.
- L'objet *Tenant* que l'on peut représenter par un projet ou une organisation au sein duquel les instances seront regroupées et administrées par les utilisateurs.
- L'objet *Rôle* qui définit le rôle de l'utilisateur sur un *Tenant*. C'est un ensemble de droits pour les utilisateurs. Un utilisateur peut avoir un ou plusieurs rôles sur différents *Tenants*.

4.2.4 Projet Nova

Nova est le cœur d'Openstack et est par conséquent l'un des composants le plus complexes. Il comporte plusieurs sous-modules ayant chacun une fonction bien précise. En effet, chaque sous-module dépend d'une base de données SQL (SQLite ou MySQL ou encore PostgreSQL) qui permet de stocker les types d'instances disponibles, les instances en cours d'exécution, etc. Ces sous-modules sont :

- Nova-api : ce programme gère les appels API de l'utilisateur. Il supporte l'API native d'Openstack ainsi que l'API EC2 d'Amazon. Il initie également le démarrage des machines virtuelles et vérifie si certaines règles sont bien respectées (quotas) ;
- Nova-compute : ce programme tourne sur les serveurs hôtes. Il gère le cycle de vie des machines virtuelles via l'API de l'hyperviseur ;
- Nova-scheduler : ce programme s'occupe de récupérer les demandes de création de machines virtuelles et de déterminer sur quelle machine hôte chaque nouvelle instance doit s'exécuter. Certes, il assure le filtrage des hôtes disponibles et la répartition des charges. Par défaut, il utilise l'algorithme Round Robin.
- Nova propose également un service permettant aux utilisateurs d'accéder à la console de leur machine virtuelle. Ce service repose sur plusieurs programmes (nova-console, nova-vncproxy et nova-consoleauth).

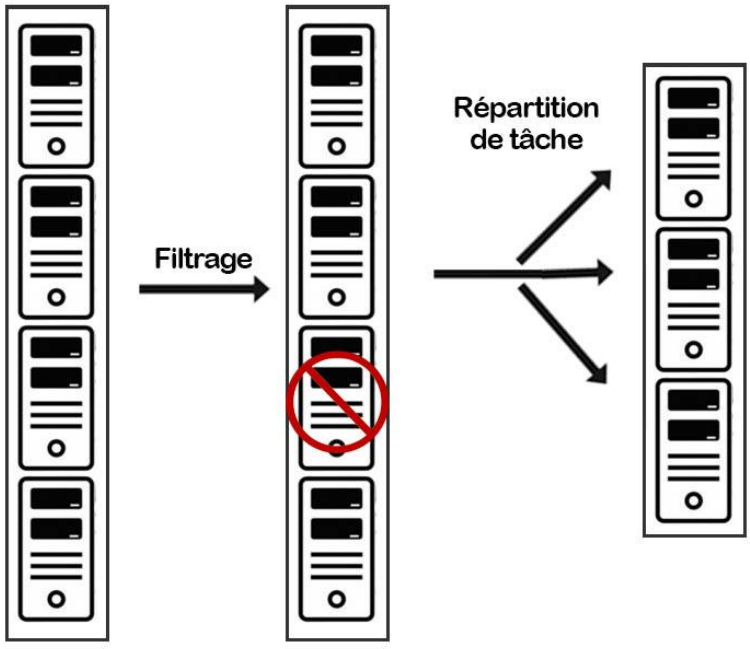

Figure 4.03: *Processus de filtrage et de répartition des charges pratiqué par nova-scheduler*

4.2.5 Projet Glance

Glance gère le catalogue d'image de disque dur de système d'exploitation utilisé par les machines virtuelles. Il assure la création, le stockage et la suppression d'une image. L'enregistrement de l'état d'une instance (snapshot) est aussi assuré par Glance. Il se compose de quatre parties principales :

- Glance-api : un programme qui traite les requêtes pour la gestion des images. Il permet notamment de lister les images disponibles, récupérer une ou en créer une nouvelle ;
- Glance-registry : un programme qui stocke, traite et récupère les métadonnées associées aux images (taille, type, etc) ;
- Base de données SQL pour stocker les métadonnées ;
- Un répertoire de stockage : pour stocker les fichiers images. Ce répertoire supporte le système de fichier normal et distribué comme Amazon S3 et Dropbox.

4.2.6 Projet Cinder

Cinder gère le stockage permanent. Pour cela, un ou plusieurs volumes de stockage sont fournis à une machine virtuelle. De plus, il permet de modifier ou supprimer les volumes déjà définis. Il repose sur 3 sous module :

- Cinder-api : programme de gestion des requêtes ;
- Cinder-volume : c'est le cœur de Cinder, il assure l'interaction avec la base de données ;
- Cinder-scheduler : comme nova-sheduler, il sélectionne le stockage le plus adapté pour le volume.

4.2.7 Projet Swift

Swift offre un service de stockage des données non structurées. On remarque qu'une structure de données est une structure logique destinée à contenir des données afin de leur appliquer une organisation permettant de simplifier leur traitement. Une structure de données implémente concrètement un type abstrait. Ce projet assure que les clients puissent stocker et récupérer beaucoup de données avec l'utilisation du protocole HTTP (GET, PUT, DELETE) et une API simple.

4.2.8 Projet Neutron

Neutron est le module gérant le réseau en tant que service (NaaS). Ce service permet aux utilisateurs de créer et gérer leur propre réseau et de se connecter à différents types d'architecture

grâce aux différents plugins (Open vSwitch, Switch Cisco, Juniper, etc). Ainsi, il offre aux clients une possibilité de créer des topologies de réseaux virtuels avancés, y compris des services tels que les pare-feu, l'équilibrage des charges et les réseaux privés virtuels (VPN).

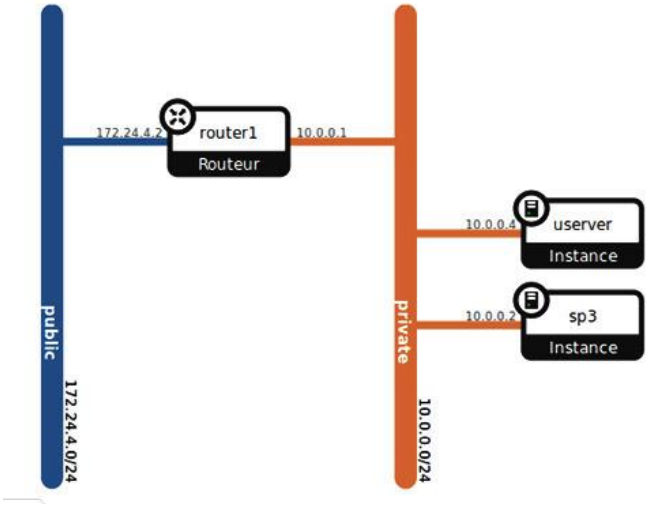

Figure 4.04: *Aperçu d'une topologie de réseau virtuel sous Neutron*

4.2.9 Projet Ceilometer

Ceilometer assure le service de métrologie dans le but d'une facturation. Dans ce cas, il collecte toutes les informations liées à la consommation en ressources de chaque instance. Tous les projets de distribution de ressource (Nova, Neutron, Swift, Glance, Cinder) contribuent donc dans la facturation en envoyant une notification au Ceilometer.

4.2.10 Interaction entre les projets

L'interaction entre les projets est assurée par le biais des messages. Par ailleurs, la synchronisation des tâches repose sur NTP (*Network Time Protocol*).

4.2.10.1 RabbitMQ (Rabbit Message Queuing)

RabbitMQ est une application open source fonctionnant comme un courtier de messages. En fait, les messages transmis sont des blocs de données binaires.

Face aux différents courtiers de messages propriétaires (MQseries, JMS, etc), la banque JPMorgan a développé le protocole AMQP (*Advanced Message Queuing Protocol*). Son objectif était de standardiser les échanges de messages entre serveurs en se basant sur les principes suivants: orienté message, utilisation de files d'attente, routage (point à point et par diffusion), fiabilité et

sécurité [22]. AMQP est donc un protocole courtier de messages inter-applicatif, ouvert pour les systèmes orientés middleware.

En effet, RabbitMQ implémentait l'AMQP en 2007 et devenait une alternative aux courtiers de message basés sur des protocoles propriétaires. Il est de nos jours utilisé par Openstack, Amazon Ec2, Google Compute Engine et d'autres solutions Clouds. De surcroit, les librairies fournies aux clients sont disponibles sous différents langages de programmation tels que Python, Java, Ruby, PHP, C#, etc. La figure suivante met en évidence les technologies prédécesseurs de RabbitMQ.

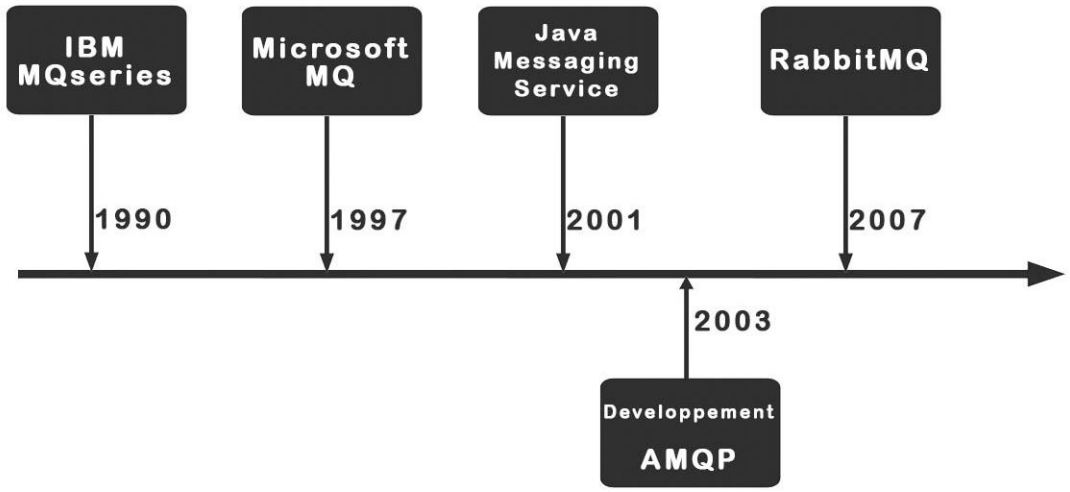

Figure 4.05: *Evolution des applications et protocole courtiers de messages [23]*

Une pile de travail forme une file d'attente pour les messages provenant des différents programmes. De ce fait, les programmes qui créent les messages sont appelés « producteurs » et ceux qui en reçoivent sont appelés « consommateurs ». Les producteurs et les consommateurs peuvent situer dans des machines différentes. Entre autres, un producteur peut envoyer un message à plusieurs producteurs et un producteur peut recevoir des messages provenant de plusieurs producteurs.

La figure suivante illustre l'interaction entre des projets d'OpenStack qu'on a expliqué précedemment.

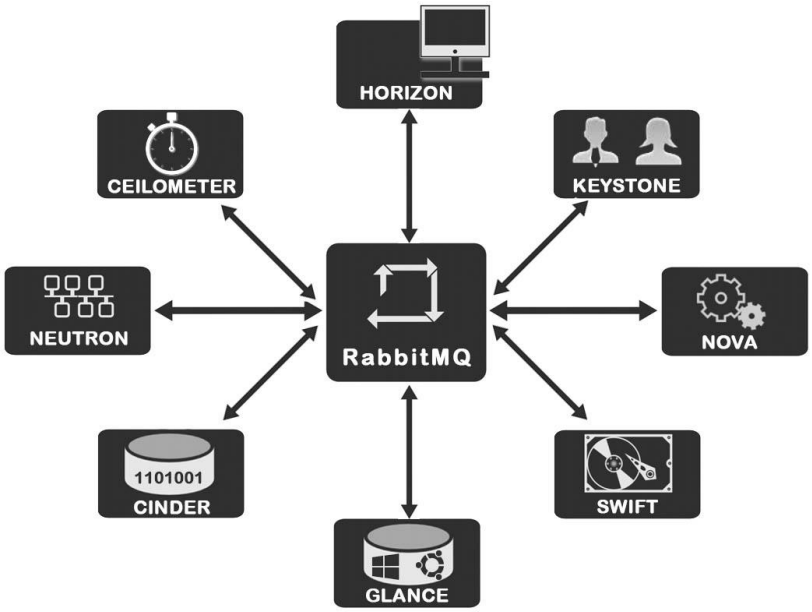

Figure 4.06: *Interaction entre les projets d'OpenStack [25]*

4.2.10.2 AMQP

AMQP est divisée en deux couches séparées. Au plus bas niveau, on rencontre un protocole Peer-to-Peer pour le transport des messages entre deux processus sur un réseau. Cela rend chaque composant d'Openstack capable d'envoyer et de recevoir des messages entre eux. Deuxièmement, en haut niveau, on rencontre un format de message abstrait, avec encodage standard. Chaque processus AMQP conforme doit être capable d'envoyer et de recevoir des messages dans ce codage standard.

Concernant le mode connexion, AMQP fonctionne en mode connecté. Chaque transmission de message doit être précédée d'un établissement de connexion et terminée par une libération du canal de communication.

4.2.10.3 NTP

L'heure des ordinateurs est gérée par une horloge à quartz. Cette technologie est fiable et la dérive est très faible, mais elle existe. Pour faciliter certains travaux d'échange entre ordinateurs, ou pour pouvoir, par exemple, exploiter les journaux (logs), les ordinateurs d'un réseau doivent être synchronisés sur la même heure.

Le protocole NTP sert à cela. Il permet aux ordinateurs de régler leur horloge sur celle d'un serveur de temps, lui-même maintenu à l'heure UTC (*Universal Time Coordinated*) selon différents moyens : connexion à une horloge atomique (qui ne dépend pas sur la rotation de la terre), à un récepteur GPS (*Global Positionning System*), ou à un autre serveur NTP.

Le travail de synchronisation se réalise par échange de messages entre serveurs de temps, avec un calcul de temps d'aller-retour des informations pour s'approcher le plus possible de l'heure officielle si bien que la transmission de l'heure se fait de façon verticale, latérale ou locale.

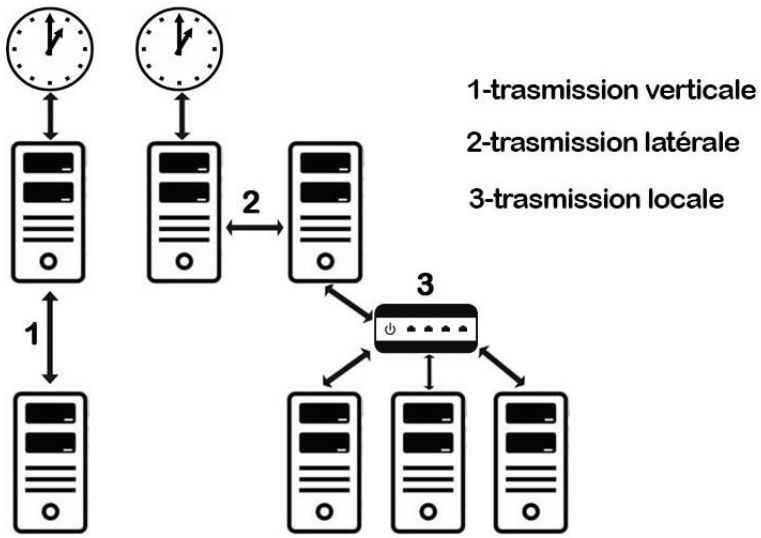

Figure 4.07: *Mode de transmission NTP*

4.3 Réalisation d'un Cloud privé à 2 nœuds

Un Cloud privé peut être réalisé à partir d'une machine physique, assez performante pour assurer le bon fonctionnement des services (IaaS, PaaS, SaaS, etc). La possibilité d'ajout d'un ou plusieurs nœuds dépend du degré de flexibilité apporté par les logiciels utilisés durant le déploiement du Cloud.

Dans le cadre de cet ouvrage, on a déployé un Cloud privé dans deux PC dont chacun tourne sur Ubuntu. La connexion entre les nœuds se fait à l'aide d'un câble pair torsadé.

Le serveur 1 est composé par 7 projets d'OpenStack et un serveur RabbitMQ. Il se comporte comme un nœud de contrôle et assure la fonctionnalité du Cloud privé déployé. Le serveur 2 est défini comme un nœud de calcul qui peut offrir sa performance en calcul aux instances invités lancés à partir du serveur 1. Certes, la mise en évidence des évaluations de performance de calcul et de stockage peut se faire sur un serveur.

L’ensemble du Cloud privé (serveurs physiques et instances) peut être représenté ainsi :

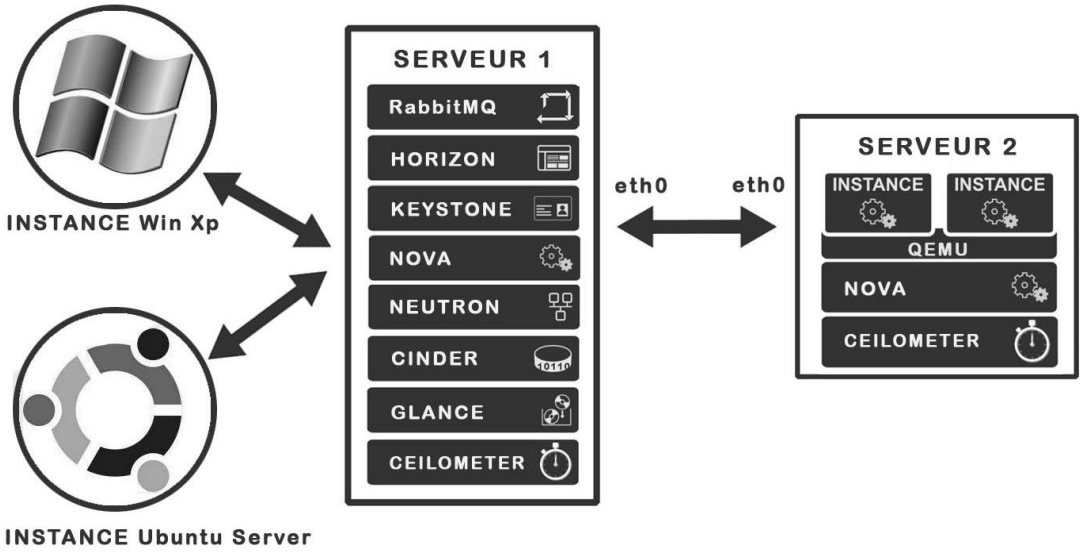

Figure 4.08: *Architecture d’un Cloud privé à base de deux nœuds*

L’installation d’OpenStack sur ces deux serveurs nécessite le téléchargement des paquets et dépendances de chaque projet. Les étapes à parcourir sont détaillées dans l’annexe 1.

Après le déploiement du Cloud privé, on a lancé deux instances basées sur Windows Xp et Ubuntu Serveur.

4.4 Evaluation de performance de calcul

4.4.1 Méthode d’évaluation

4.4.1.1 Désactivation des programmes inutiles

Le test de performance doit être réalisé dans un système qui ne partage pas ses ressources à d’autres programmes inutiles à son fonctionnement. Quoiqu’ils soient inaperçus, les programmes en arrière-plan consomment des ressources, leurs démarrages peuvent être simultanés à celui du système. La liste de chaque processus en cours d’exécution est visible dans le gestionnaire de tâche, c’est pareil pour Windows et Linux.

4.4.1.2 Détermination de la dimension maximale du système d’équation

La capacité mémoire requise pour un système d’équation de dimension $n \times n$ est obtenue à partir de l’équation (3.09).

Dans notre cas, le nombre de bits servant à coder un nombre flottant est de 64bits ou 8octets. La dimension maximale du système d'équation à résoudre est donc donnée par la formule suivante.

$$D_{sys} = E\left[\frac{1}{2}\sqrt{\frac{M_{disp}}{2}}\right] \qquad (4.01)$$

$E[x]$: partie entière de x;

D_{sys}: dimension du système d'équation ;

M_{disp} : mémoire vive du système disponible.

L'utilisation de la totalité de la mémoire vive du système peut confirmer la bonne fonction de chaque unité de RAM. Certes, cette application nécessite beaucoup de temps du fait que la taille de la mémoire vive utilisée est proportionnelle à la dimension du système d'équation à résoudre. La performance maximale n'est pas forcément obtenue durant la résolution d'un gros système d'équations.

4.4.1.3 Lancement de plusieurs tests et variation de la dimension du système d'équation

Durant le test, on commence par une petite dimension jusqu'à la dimension maximale si nécessaire. De plus, la moyenne est calculée pour chaque dimension et notée par « $\bar{P}$ ».

L'apparition d'une valeur petite par rapport aux autres, dans la catégorie d'une dimension, marque une anomalie. Cette valeur doit être changée par une autre valeur de test pour que la performance moyenne ne soit pas faussée.

27.87	15.69	27.01	27.59	28.04	14.87	27.94	27.7	27.84

Tableau 4.01: *Anomalies durant un test sur une dimension d'équation*

Le problème se pose sur le point où on doit arrêter le test. D'après plusieurs évaluations effectuées, la courbe formée par la puissance moyenne se rapproche vite d'une droite horizontale qu'elle suit jusqu'au test sur une dimension maximale du système d'équations. Dans ce cas, l'arrêt du test peut être marqué par une apparition de quelques performances moyennes successives de même valeur avec une grande différence entre deux dimensions successives.

La figure suivante représente l'évolution de la performance de calcul du serveur 1.

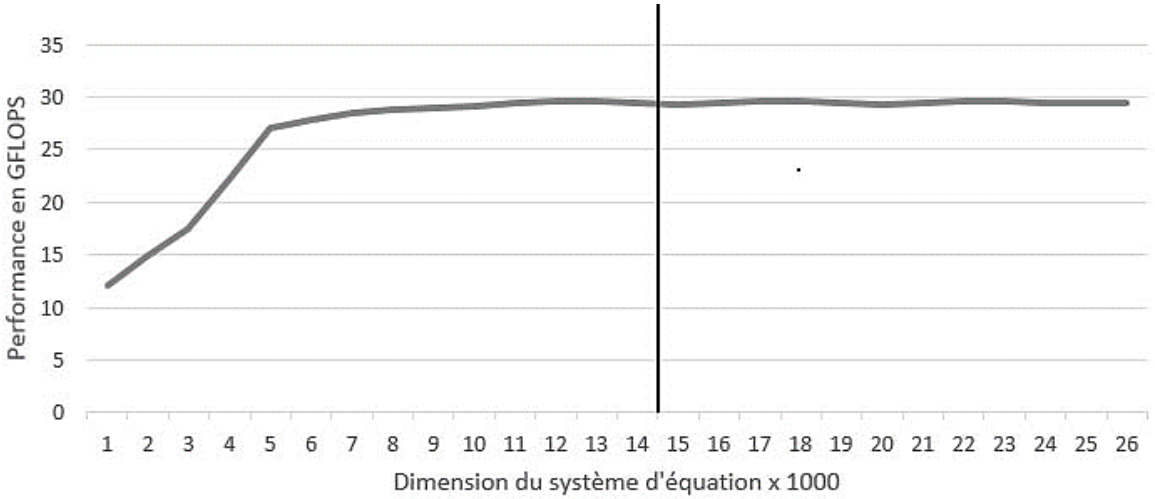

Figure 4.09: *Performance de calcul selon la dimension du système d'équation à résoudre*

Dans la suite, le point d'arrêt est défini par l'apparition de trois valeurs identiques (10^{-1} près) de la performance de calcul moyenne.

4.4.2 Evaluation du serveur 1 sous Ubuntu

4.4.2.1 Caractéristique

Ordinateur portable :

- Marque : ASUS ;
- CPU : Intel Core i3-2310M 2,1GHz x 4 $\Leftrightarrow P_{théorique} = 33{,}6\ GFLOPS$;
- RAM : 6Go dont 5,1Go disponible ;
- HDD : 500Go ;
- OS : Ubuntu_desktop 14.04 .

Dimension maximale du système d'équations à résoudre :

$$D_{sys} = E\left[\frac{1}{2}\sqrt{\frac{5{,}1 \times 2^{30}}{2}}\right] \tag{4.02}$$

$$D_{sys} = 26163 \tag{4.03}$$

4.4.2.2 Résultat de l'évaluation de performance de calcul

Test / Dimension	1	2	3	4	5	6	7	8	9	10	$\bar{P}$
1000	14.01	13.98	13.95	14.02	15	14.06	11.61	11.79	11.8	12.03	13.23
2000	15.39	18.4	16.87	15.67	13.38	13.32	16.01	15.93	15.19	15.02	15.52
3000	12.95	17.26	17.65	19.45	19.76	17.25	18.1	19.81	16.72	17.57	17.65
4000	22.75	24.58	25.36	24.95	25.08	25.2	25.32	25.92	25.51	22.26	24.69
5000	27.03	26.85	26.8	27.29	26.74	27.41	26.88	27.25	26.85	27.06	27.02
6000	27.74	27.87	27.69	27.01	27.59	28.04	27.87	27.94	27.7	27.84	27.73
7000	28.25	28.33	28.52	28.3	28.36	28.32	28.33	28.44	28.58	28.42	28.39
8000	25.19	27.97	28.71	28.72	28.78	28.89	28.72	28.86	28.82	28.84	28.35
9000	28.65	28.89	28.67	28.87	28.92	28.86	28.46	28.74	28.8	28.9	28.78
10000	29.19	29.31	29.15	28.98	29.07	29.33	29.18	29.24	29.38	29.07	29.19
11000	29.55	29.56	29.26	29.18	29.55	29.55	29.5	29.48	29.32	29.44	29.44
12000	29.57	29.62	29.66	29.66	29.67	29.7	29.46	29.62	29.59	29.68	29.62
13000	29.52	29.43	29.68	29.32	29.65	29.71	29.64	29.64	29.32	29.68	29.56
14000	29.66	29.66	29.67	29.42	29.52	29.42	29.55	29.46	29.53	29.48	29.54

Tableau 4.02: *Performance du serveur 1 en FLOPS*

La présence de la valeur de la moyenne colorée en vert marque l'arrêt de l'évaluation au lieu d'une poursuite jusqu'à la dimension 26000. Ainsi, la performance maximale (P_{max}) du serveur 1 est égale à 29,71 FLOPS.

4.4.3 Instance basée sur Ubuntu Serveur

4.4.3.1 Caractéristique

Machine virtuelle lancée sur le serveur 1 :

- CPU : 1 CPU ;
- RAM : 1Go dont 950Mo disponible ;
- HDD : 5Go ;
- OS : Ubuntu_serveur 14.04.

Dimension maximale du système d'équation à résoudre :

$$D_{sys} = E\left[\frac{1}{2}\sqrt{\frac{950 \times 2^{20}}{2}}\right] \quad (4.04)$$

$$D_{sys} = 11158 \quad (4.05)$$

4.4.3.2 Résultat de l'évaluation de performance de calcul

Test / Dimension	1	2	3	4	5	6	7	8	9	10	$\bar{P}$
1000	3.21	3.68	3.91	3.66	3.80	4.15	3.68	4.85	4.98	4.20	4.01
2000	4.23	4.78	5.23	4.93	4.68	4.90	5.11	5.11	4.09	4.90	4.80
3000	4.60	5.20	4.97	4.98	5.14	4.81	5.21	5.27	4.88	5.20	5.03
4000	5.01	4.99	5.05	5.16	5.05	5.25	4.89	5.21	5.16	4.99	5.08
5000	5.11	5.10	5.34	5.28	5.10	5.21	5.27	5.18	5.16	5.12	5.19
6000	5.49	5.39	5.43	5.66	5.64	5.58	5.67	5.69	5.59	5.53	5.57
7000	5.50	5.69	5.50	5.53	5.54	5.55	5.62	5.52	5.65	5.48	5.56
8000	5.50	5.44	5.51	5.51	5.62	5.61	5.53	5.44	5.51	5.53	5.52

Tableau 4.03: *Performance de calcul de l'instance Ubuntu Serveur en FLOPS*

$$P_{max} = 5{,}69 \text{ FLOPS} \quad (4.06)$$

4.4.4 Instance basée sur Windows Xp

4.4.4.1 Caractéristique

Machine virtuelle lancée sur le serveur 1 :

- CPU : 1 CPU ;
- RAM : 1Go dont 920Mo disponible ;
- HDD : 5Go ;
- OS : Windows XP-SP3.

Dimension maximale du système d'équations à résoudre :

$$D_{sys} = E\left[\frac{1}{2}\sqrt{\frac{920 \times 2^{20}}{2}}\right] \quad (4.07)$$

$$D_{sys} = 10981 \quad (4.08)$$

4.4.4.2 Résultat de l'évaluation de performance de calcul

Test / Dimension	1	2	3	4	5	6	7	8	9	10	$\bar{P}$
1000	3.12	3.68	3.92	3.77	4.06	3.83	4.10	3.99	3.94	4.12	3.85
2000	4.90	5.11	5.09	4.94	4.96	5.04	4.69	5.13	4.78	4.91	4.96
3000	5.19	5.29	5.26	4.99	5.22	5.22	5.18	5.13	4.97	5.24	5.17
4000	5.21	5.19	5.12	5.12	5.23	5.08	5.07	5.12	5.06	5.10	5.13
5000	5.11	5.18	5.16	5.25	5.17	5.12	5.20	5.19	5.17	5.21	5.18
6000	5.16	5.28	5.20	5.02	5.18	5.14	5.13	5.19	5.20	5.17	5.17
7000	5.17	5.19	5.03	5.10	5.24	5.16	5.18	5.09	5.11	5.20	5.15

Tableau 4.04: *Performance de calcul de l'instance Windows XP-SP3 en FLOPS*

$$P_{max} = 5{,}29\ GFLOPS \quad (4.09)$$

4.4.5 Interprétation des résultats

Pour le serveur 1, qui en a 4 CPU, la performance en calcul maximale est de 29,71 FLOPS, c'est-à-dire, 7,43 GFLOPS par CPU. De plus, l'architecture Cloud déployée dans le serveur permet la création de 4 instances dotées d'un CPU virtuel chacun. L'évaluation de la performance en calcul de l'instance basée sur Windows Xp donne 5,29 GFLOPS et celle de l'instance basée sur Ubuntu Serveur donne 5,69 GFLOPS. La différence de 0,5 GFLOPS est due à l'utilisation d'une interface graphique sur le système d'exploitation Windows Xp.

Dans le domaine de virtualisation, le rendement est obtenu à partir de la performance en calcul de la machine hôte et la machine virtuelle. Cela implique que le rendement de virtualisation dépend de l'hyperviseur et le système d'exploitation de la machine invitée. L'hyperviseur utilisé durant la réalisation est QEMU.

On peut déduire donc les rendements de virtualisation suivants :

- Rendement de virtualisation de QEMU avec Windows Xp :

$$R_{QEMU,\ WIn\,XP} = \frac{5,29}{7,43} \times 100 \tag{4.10}$$

$$R_{QEMU,\ WIn\,XP} = 71\% \tag{4.11}$$

- Rendement de virtualisation de QEMU avec Ubuntu Server :

$$R_{QEMU,\ U\,Serveur} = \frac{5,69}{7,43} \times 100 \tag{4.12}$$

$$R_{QEMU,\ U\,Serveur} = 76\% \tag{4.13}$$

On peut donc généraliser que le rendement de QEMU est de l'ordre de 70%. Cela confirme aussi que l'hyperviseur, implémenté dans le projet Nova joue un rôle très important sur la performance des instances. Entre autres, le changement de QEMU par un autre hyperviseur pourra améliorer ou dégrader la performance en calcul des instances.

4.4.6 Evaluation de performance de stockage

4.4.6.1 Méthode d'évaluation

Comme dans le cas du test de performance de calcul, le test de performance de stockage doit être réalisé dans un système qui ne partage pas ses ressources à d'autres programmes inutiles à son fonctionnement.

Le test se fait dans une partie du disque de stockage, le volume maximal est défini par l'espace libre dans ce dernier. HD Tune à limiter le volume minimal à 50Mo.

L'évaluation de la performance de stockage est basée sur un lancement de plusieurs tests et variation du volume du disque à évaluer. Chaque évaluation élémentaire tente de déterminer ou a pour but de déterminer le débit de transfert sur la lecture/écriture séquentielle et la performance en entrée/sortie (en IOPS) pour un bloc de fichier multiple de 4Ko.

Durant l'évaluation, on a retenu la vitesse en lecture/écriture séquentielle et la performance en entrée/sortie (en IOPS) pour un bloc de fichier de 4Ko. Selon plusieurs tests, les résultats obtenus sur un volume minimal jusqu'au volume maximal se trouvent toujours près d'une droite qui

représente la performance moyenne. En somme, le critère d'arrêt n'est plus basé sur la moyenne des tests effectués. On a arrêté l'évaluation après quelques tests, car les résultats se trouvent déjà sur le voisinage de la moyenne.

4.4.6.2 Evaluation du serveur 1

Volume(Mo)	50	150	250	350	450	550	650	750	850	950	$\bar{x}$
V_{LS} (Ko/s)	51543	51220	50739	47667	46736	47522	44667	44419	45463	47233	47721
V_{ES} (Ko/s)	50436	48766	42888	46643	45347	46377	44450	46120	44256	46915	46220
S_{IOPS}	99	100	98	105	112	115	103	107	105	103	105
E_{IOPS}	259	235	230	223	203	206	192	204	198	202	215

Tableau 4.05: *Performance de stockage du serveur 1*

V_{LS} : vitesse de lecture séquentielle ;

V_{ES} : vitesse d'écriture séquentielle ;

S_{IOPS} : nombre d'opération de sortie par seconde ;

E_{IOPS} : nombre d'opération d'entrée par seconde.

La performance de stockage du serveur 1 peut être caractérisée par les valeurs moyennes suivantes :

- $\overline{V_{LS}} = 47721\ Ko/s$;
- $\overline{V_{ES}} = 46220\ Ko/s$;
- $\overline{S_{IOPS}} = 105\ IOPS$;
- $\overline{E_{IOPS}} = 215\ IOPS$.

4.4.6.3 Evaluation de l'instance Windows Xp

Volume(Mo)	50	150	250	350	450	550	650	750	850	950	$\bar{x}$
V_{LS} (Ko/s)	26531	29151	33944	34295	28034	36102	35871	30326	33102	35322	26531
V_{ES} (Ko/s)	16232	13663	15100	17277	16453	16566	15148	18750	15691	16782	16232
S_{IOPS}	87	82	77	67	80	79	75	77	58	64	75
E_{IOPS}	59	61	59	89	52	79	64	75	54	90	68

Tableau 4.06: *Performance de stockage de l'instance Windows Xp-Sp3*

La performance de stockage de l'instance Windows Xp peut être caractérisée par les valeurs moyennes suivantes :

- $\overline{V_{LS}} = 26531\ Ko/s$;
- $\overline{V_{ES}} = 16232\ Ko/s$;
- $\overline{S_{IOPS}} = 75\ IOPS;$
- $\overline{E_{IOPS}} = 68\ IOPS.$

4.4.6.4 Interprétation des résultats

Sur le serveur 1, on constate que les vitesses de lecture et d'écriture séquentielle ont presque la même valeur. Par contre, le nombre d'opérations de sortie par seconde est environ la moitié du nombre d'opération d'entrée par seconde. Pour la validation de ces proportions, on a évalué un autre disque de stockage extérieur et on a abouti à la même proportion. On en déduit qu'il y a plusieurs facteurs qui affectent les résultats IOPS, y compris la configuration du système, des pilotes de stockage, les opérations d'arrière-plan du système d'exploitation, etc.

A propos du rendement de virtualisation, on a les résultats suivants :

- $R_{\overline{V_{LS}}} = 55\%;$
- $R_{\overline{V_{ES}}} = 35\%;$
- $R_{\overline{S_{IOPS}}} = 71\%;$
- $R_{\overline{E_{IOPS}}} = 31\%.$

Suivant ces résultats, on en déduit que le rendement de la performance de stockage dépasse 31% et d'autres rendements dépassent 50%. Par rapport au rendement sur la performance de calcul, les

projets d'OpenStack, responsables des images des systèmes virtuels et volumes de stockage de données ont plus besoin d'optimisation.

4.5 Conclusion

Finalement, on peut dire qu'OpenStack permet le déploiement d'un Cloud privé sur un serveur physique avec une possibilité d'ajout de plusieurs nœuds. Le Cloud privé déployé à partir de ce dernier a un rendement élevé (supérieur à 70%) par rapport à la performance de calcul du serveur physique. Par ailleurs, la performance de stockage des instances est faible, et le rendement de 31 à 71% par rapport à la performance de stockage du serveur physique.

CONCLUSION GENERALE

L'évolution de la technologie a permis l'assemblage de plusieurs serveurs en un seul système informatique. Le terme Cloud apparait après l'application d'une abstraction des matériels physiques aux clients, qui bénéficient des services à la demande. On a pu classer 3 modes de déploiement de Cloud et 3 types de services standard. Dans tous les cas, les outils sélectionnés (LINPACK et HD Tune) ont l'aptitude de mesurer la performance de calcul et la performance de stockage des systèmes informatiques dont les résultats sont donnés selon des unités standardisées. Les différentes possibilités de test présenté par ces outils permettent en fait l'évaluation de performance des systèmes informatiques très faible jusqu'à l'évaluation de performance des superordinateurs.

Le présent travail a évoqué l'architecture détaillée d'OpenStack, pratiquée dans un déploiement d'un Cloud privé. Nous l'avons focalisé sur l'évaluation de la performance de la partie responsable de calcul et de stockage. La connaissance de l'architecture d'un processeur et des disques de stockages (HDD et SSD) a mené vers la compréhension du système informatique étudié. Les résultats issus de LINPACK montrent clairement que l'hyperviseur utilisé en a une rentabilité supérieure à 70% dans le cadre de la distribution des ressources de calcul. Par ailleurs, le rendement de la distribution des ressources en stockage évalué par HD Tune est cadré entre 35 et 71%.

On peut en dire donc que le but est atteint, les résultats obtenus nous attirent vers le classement de chaque technologie utilisée par OpenStack selon leur performance. Entre autres, la voie vers le domaine d'optimisation est ouverte. Le changement des parties d'OpenStack par d'autres est aussi possible à condition que ces derniers représentent un rendement élevé.

ANNEXES

ANNEXE 1 DEPLOIEMENT D'UN CLOUD PRIVE AVEC OPENSTACK

Présentation

La documentation officielle d'Openstack montre pas à pas le mode de déploiement. C'est une installation par projet. Toutefois, une méthode d'installation par script est possible, les démarches et paramétrages réalisés pendant l'élaboration de cet ouvrage seront représentées ci-dessous.

Démarche

L'installation d'OpenStack requiert une récupération des données hébergées par « github.com ». Pour se faire, on doit installer premièrement le paquet « git » (commande console : sudo apt-get install git).

Dans le cas de ce présent ouvrage, on a installé la version « icehouse ». On peut télécharger seulement la branche contenant l'installation de « icehouse » par la commande suivant :

git clone -b stable/icehouse https://github.com/openstack-dev/devstack.git

Après le téléchargement, on devrait avoir un dossier nommé « devstack ». Ce dossier contient des scripts utiles pour le lancement de l'installation et la désinstallation. Ces fichiers sont respectivement nommés « stack.sh » et « unstack.sh ». Certes, le lancement immédiat de ces scriptes n'est pas possible à moins qu'on leur donne un attribue exécutable. Cela se fait par la commande suivant :

cd devstack

chmod 777 -R devstack/

Si tout marche bien, il ne reste qu'une démarche à suivre avant le lancement du script d'installation. C'est la création d'un fichier nommé « localrc », utilisé comme fichier de paramétrage.

Fichier de paramétrage

Ce fichier contient une indication pour le script « stack.sh » tels que :

- Le nom des projets à installer ;
- Les paramètres de chaque projets (ex pour keystone : nom d'utilisateur, mot de passe) ;
- Adresse IP
- …

Fichier de paramétrage du serveur 1

```
DEST=/opt/stack
# Logging
LOGFILE=$DEST/logs/stack.sh.log
VERBOSE=True
OFFLINE=True
LOG_COLOR=False
SCREEN_LOGDIR=$DEST/logs/screen

# Credentials
ADMIN_PASSWORD=openstack
MYSQL_PASSWORD=openstack
RABBIT_PASSWORD=openstack
SERVICE_PASSWORD=openstack
SERVICE_TOKEN=tokentoken

# Github's Branch
GLANCE_BRANCH=stable/icehouse
HORIZON_BRANCH=stable/icehouse
KEYSTONE_BRANCH=stable/icehouse
NOVA_BRANCH=stable/icehouse
NEUTRON_BRANCH=stable/icehouse
HEAT_BRANCH=stable/icehouse
CEILOMETER_BRANCH=stable/icehouse

# HOST
#EDITME
HOST_IP=192.168.56.109

#Services
disable_service n-net
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
```

```
enable_service neutron

# Heat - Orchestration Service
ENABLED_SERVICES+=,heat,h-api,h-api-cfn,h-api-cw,h-eng
HEAT_STANDALONE=True

# Ceilometer - Metering Service (metering + alarming)
ENABLED_SERVICES+=,ceilometer-acompute,ceilometer-acentral,ceilometer-
collector,ceilometer-api
ENABLED_SERVICES+=,ceilometer-alarm-notify,ceilometer-alarm-eval

#Scheduler
SCHEDULER=nova.scheduler.chance.ChanceScheduler
```

Fichier de paramétra du serveur 2

```
DEST=/opt/stack

# Logging
LOGFILE=$DEST/logs/stack.sh.log
VERBOSE=True
LOG_COLOR=False
SCREEN_LOGDIR=$DEST/logs/screen
OFFLINE=true

# Credentials
ADMIN_PASSWORD=openstack
MYSQL_PASSWORD=openstack
RABBIT_PASSWORD=openstack
SERVICE_PASSWORD=openstack
SERVICE_TOKEN=tokentoken

# Host detail
#EDITME
HOST_IP=192.168.56.108
#EDITME
SERVICE_HOST=192.168.56.109
Q_HOST=$SERVICE_HOST

# Github's Branch
GLANCE_BRANCH=stable/icehouse
HORIZON_BRANCH=stable/icehouse
KEYSTONE_BRANCH=stable/icehouse
NOVA_BRANCH=stable/icehouse
NEUTRON_BRANCH=stable/icehouse
HEAT_BRANCH=stable/icehouse
CEILOMETER_BRANCH=stable/icehouse
```

```
# Services
disable_all_services
ENABLED_SERVICES=neutron,n-cpu,rabbit,q-api,q-agt
```

ANNEXE 2 CREATION D'IMAGE POUR LES INSTANCES

La création d'image pour les instances se divise en trois parties :

- Création de l'image proprement dite
- Installation d'un système d'exploitation
- Exportation de l'image vers le projet Glance

Création d'une image

On peut utiliser la commande « qemu-img create » pour créer une image. Par ailleurs, on doit spécifier dès la création d'une image sa taille et son format. La commande suivante crée une image de 10G sous un format QCOW2 (QEMU Copy on Write version 2) qui est propre à QEMU.

- qemu-img create -f qcow2 winxpsp3.qcow2 10G

Installation d'un système d' exploitation

L'installation d'un système d'exploitation nécessite un fichier copie de l'installation sous un format ISO. La commande suivant lance une machine virtuelle ayant winxpsp3.qcow2 comme HDD et cdvirtualfile.iso comme CDROM.

- #kvm -hda winxpsp3.qcow2 \
- -drive file=winxpsp3.qcow2,if=virtio \
- -drive file=cdvirtualfile.iso,media=cdrom,index=1 \
- -net nic,model=virtio \
- -net user \
- -boot d \
- -vga std \
- -m 1024

Après le lancement, l'installation se déroule comme les installations sur une machine physique.

Exportation de l'image vers le projet Glance

L'exportation de l'image vers le projet Glance peut se faire en ligne de commande ou en utilisant directement l'interface fournie par le projet Horizon.

BIBLIOGRAPHIE

[1] T. Wigwam, *« Le Cloud Computing : Réelle évolution ou simple évolution »*, Bureau de centre d'expertise technologique

[2] R. Elino, « *Développement d'Applications d'Entreprise* », Cours I5-TCO, Dpt Télécommunications ESPA 2014

[3] J. Bolot., W. Dabbous, « *L'Internet: Historique et évolution*», INRIA Sophia Antipolis, 2004

[4] S. Krakowiak, «*Introduction aux systèmes et applications répartis* », Projet Sardes (INRIA et IMAG-LSR), Université Joseph Fourier, Octobre 2006.

[5] A. Safidy, « *Optimisation de performance d'une grille de calcul* », ESPA, 2014

[6] K. Hess, A. Newman, « *La vitualisation en pratique* », Pearson Education, 2010

[7] C. Sylvain, S. Georges, « *Guide Complet Cloud Computing* », MA Éditions

[8] *http://www.salesforce.com/fr/socialsuccess/cloud-computing*, 2015

[9] A. Velte., R. Elsenpeter, « *Cloudcomputing: a practical approach* », Interactive week, 2010

[10] A. Petitet, R. Whaley, J. Dongarra, A. Cleary, *« HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers* », University of Tennesse, sep 2008

[11] http://www.top500.org/statistics/sublist/, 2015

[12] M. GILLI, « *Méthodes numériques* », Université de Genève, mars 2006

[13] « *HD Tune Pro manual* », EFD Software, 2013

[14] « *Get S.M.A.R.T. for Reliability* », Seagate Technology, 1999

[15] H. Arpaci, C. Arpaci, « *Operating systems* », university of wisconsin, 2014

[16] M. Cédric, « *Informatique et réseaux* », Ecole nationale supérieure d'ingénieurs de Caen

[17] M. Polte, J. Simsa, G. Gibson « *Comparing Performance of Solid State Devices and Mechanical Disks* », Carnegie Mellon University

[18] C. Hossain, « *How Flash Memory Works* », Stanford University, 2012

[19] C. Henderson, « *Building scalable web sites* », O'Reilly Media, 2006

[20] J. Horwitz, *« Unix system management: primer plus »*, Sams Publishing, 2002

[21] « OpenStack Installation Guide for Ubuntu 12.04/14.04 (LTS) », OpenStack Foundation, 2014

[22] A. Clapaud, « *AMQP rajeunit la messagerie interapplicative* », 2008

[23] A. Videla, J. Williams, « *RabbitMQ in action* », MEAP Edition, 2011

[24] Y. Parent, M. Lemaux, « *Cloud Computing* », Université Nancy 2, 2011

[25] A. Jha, D. Johnson, « *OpenStack Beginner's Guide* », 2012

FICHE DE RENSEIGNEMENTS

Nom : RAMBELOSON

Prénoms : Andriantomefisoa

Adresse de l'auteur : CSB2 Sabotsinamatoana
Ambatolampy 104 – Madagascar
Tel : +261 33 12 933 77
E-mail : r.tomefy@gmail.com

Titre du mémoire :

« EVALUATION DE PERFORMANCE DE CALCUL

ET DE STOCKAGE D'UN CLOUD PRIVE»

Nombre de pages : 80

Nombre de tableaux : 9

Nombres de figures : 40

Directeur de mémoire :

Nom : BOTO ANDRIANANDRASANA

Prénoms : Jean Espérant

Grade : Assistant d'Enseignement et de Recherche

Tel : +261 34 04 281 43

E-mail : jean.respira@gmail.com

RESUME

L'évolution de la technologie et les besoins de chaque individu et entreprise mènent le monde entier vers l'ère du Cloud Computing. Cette cinquième génération de l'architecture informatique définit la façon dont on partage les ressources d'un ou plusieurs serveurs physiques sous forme de service. Les clients bénéficiaires de ces services ne chassent point l'architecture réelle des systèmes informatiques du fournisseur. Certes, des outils tels que LINPACK et HD Tune peuvent évaluer respectivement la performance de calcul et la puissance de stockage des systèmes informatiques réels et virtuels. Le déploiement d'un Cloud privé requiert aussi un outil comme OpenStack, qui est un ensemble de plusieurs projets, initialement développé par la NAZA et Rackspace. Le déploiement d'un Cloud privé et l'évaluation de la performance de calcul et de stockage montrent qu'OpenStack, avec l'utilisation de QEMU a un rendement de 70% sur le domaine de calcul. Quant aux modules responsables des unités de stockage, ils ont un rendement de 35 à 71%.

Mots clés : cloud privé, performance, LINPACK, HD Tune, OpenStack.

ABSTRACT

The evolution of technology and the needs of each individual and company lead the world towards the Cloud Computing age. This fifth generation of computer architecture defines the way in which resources of one or more physical servers are shared as a service. Customers benefit from these services do not know the real architecture of the supplier's computer systems. Therefore, tools such as LINPACK and HD Tune can each evaluate performance computing and performance storage of real and virtual IT systems. Deploying a private Cloud requires a tool like OpenStack, which is a set of several projects, originally developed by Naza and Rackspace. The deployment of private cloud and the evaluation of performance computing and storage show that OpenStack, with the use of QEMU has 70% of return in the computational domain. As for the modules of the storage units, they have 35 to 71% of return.

Keywords: private cloud, performance, LINPACK, HD Tune, OpenStack.