

UNIVERSITE MOHAMMED V DE RABAT

ECOLE MOHAMMADIA D'INGENIEURS



Génie Informatique

CONCEPTION D'UN SYSTÈME  
DISTRIBUÉ : ÉTUDE DE CAS

**Réalisé par :**

ETTAHI Yahya

&

METRANE Abdelmounim

Année universitaire 2019/2020

# Sommaire :

<b>I.</b>	<b>Introduction :</b> .....	<b>4</b>
<b>A.</b>	<b>Contexte :</b> .....	<b>4</b>
<b>B.</b>	<b>Besoin d'une application distribué :</b> .....	<b>4</b>
<b>II.</b>	<b>Architecture globale et philosophie de conception :</b> .....	<b>5</b>
<b>A.</b>	<b>Modèle physique :</b> .....	<b>5</b>
<b>B.</b>	<b>Architecture globale du système :</b> .....	<b>6</b>
<b>1.</b>	<b>les paradigmes de communication :</b> .....	<b>9</b>
<b>2.</b>	<b>Collection des données :</b> .....	<b>9</b>
<b>3.</b>	<b>Traitement des données :</b> .....	<b>9</b>
<b>4.</b>	<b>Visualisation des données :</b> .....	<b>9</b>
<b>III.</b>	<b>Conclusion :</b> .....	<b>10</b>

## Table de figures :

Figure 1 : Architecture de centre de données.....	6
Figure 2 : L'architecture globale du système.....	7
Figure 3 : Infrastructure du système. ....	8

# I. Introduction :

On voit que les organisations se sont intéressées à la décentralisation du traitement tout en réalisant l'intégration des ressources d'information au sein de leurs systèmes géographiquement répartis de base de données, applications et utilisateurs. Les systèmes et applications Web sont devenus des outils informatiques sophistiqués fournissent une fonction autonome aux utilisateurs.

## A. Contexte :

L'objectif est de proposer un système avec tous ses éléments et qui permet de collecter, traiter et visualiser des données issues de tous les hôpitaux publics du Maroc et ce quelques soit leur types (Dispensaires, hôpital régional ou CHU).

Le système doit manipuler (collecte, traitement et visualisation) deux types de données : les données opérationnelles et les données patients.

## B. Besoin d'une application distribué :

Les besoins suivant nous poussent vers un système distribué.

**Synchronisation** : assurer la collaboration, la synchronisation entre processus répartis et la coordination pour les valeurs partagées, ceci est nécessaire pour que le système permet la collecte de ses données, y effectuer des traitements préliminaires et de les mettre à la disposition des décideurs à différents niveaux : directeurs des hôpitaux concernés, les directeurs de santé régionaux et le directeur central au ministère.

**Communication** : Gérer l'échange d'information et de donnée entre différentes machines, ainsi la mettre ces données à la disposition de certains centres/laboratoires de recherche médicaux afin d'effectuer des simulations et des traitements de recherche scientifique.

**Cohérence & Réplication** : améliorer la fiabilité et la performance, pour y effectuer plusieurs requêtes sur ces données pour calculer différents indicateurs et faire d'autres des traitements préliminaires et de les mettre à la disposition des décideurs afin de prendre les décisions nécessaires.

**Tolérance aux fautes** : effectuer le recouvrement en cas de panne partielle .

**Virtualisation** : réduire la multiplicité des plateformes, adapter les interfaces et offrir un degré élevé de portabilité et de flexibilité.

## II. Architecture globale et philosophie de conception :

On adoptera comme architecture, l'architecture Client/serveur et en guise de performance du système en termes de temps d'exécution de traitement, on va choisir comme type de système distribué : un système de traitement distribué basé sur la parallélisation des clusters.

### A. Modèle physique :

Un modèle physique est une représentation des éléments matériels sous-jacents d'un système distribué qui s'éloigne des détails spécifiques des technologies informatiques et de mise en réseau utilisées.

Le choix en termes d'infrastructure physique consiste à utiliser un très grand nombre d'ordinateurs de base pour produire un environnement rentable pour le stockage distribué et le calcul. Un ordinateur donné aura généralement environ 2 téraoctets de stockage sur disque et environ 32 gigaoctets de DRAM (mémoire vive dynamique) et exécutera une version réduite du noyau Linux.

Nous allons concevoir l'infrastructure en utilisant une gamme de stratégies pour tolérer quelques défaillances. (dues aux logiciels ainsi que matérielles)

L'architecture physique est construite comme suit :

- Les PC de base sont organisés en racks avec entre 40 et 80 PC dans un rack donné. Les racks sont double face avec la moitié des PC de chaque côté. Chaque rack possède un commutateur Ethernet qui fournit une connectivité à travers le rack et également au monde externe (voir ci-dessous). Ce commutateur est modulaire, organisé en plusieurs lames, chaque lame prenant en charge 8 interfaces réseau 100 Mbits / s ou une seule interface 1 Gbit / s. Pour 40 PC, cinq lames contenant chacune huit interfaces réseau sont suffisantes pour assurer la connectivité au sein du rack. Deux autres lames, chacune prenant en charge une interface réseau à 1 Gbit / s, sont utilisées pour la connexion au monde extérieur.
- Les racks sont organisés en grappes, qui sont une unité clé de gestion, déterminant par exemple le placement et la réplication des services. Un cluster se compose généralement de 30 racks ou plus et de deux commutateurs à large bande passant fournissant une connectivité au monde extérieur (Internet et autres services). Chaque rack est connecté aux deux commutateurs pour la redondance; en outre, pour plus de redondance, chaque commutateur dispose de liaisons redondantes avec le monde extérieur.
- Les clusters sont hébergés dans des centres de données répartis dans le Maroc entier.

Une vue simplifiée de cette organisation globale est fournie à la figure suivante. Cette infrastructure physique nous fournit d'énormes capacités de stockage et de calcul, ainsi que la redondance nécessaire pour créer des systèmes à grande échelle tolérants aux pannes.

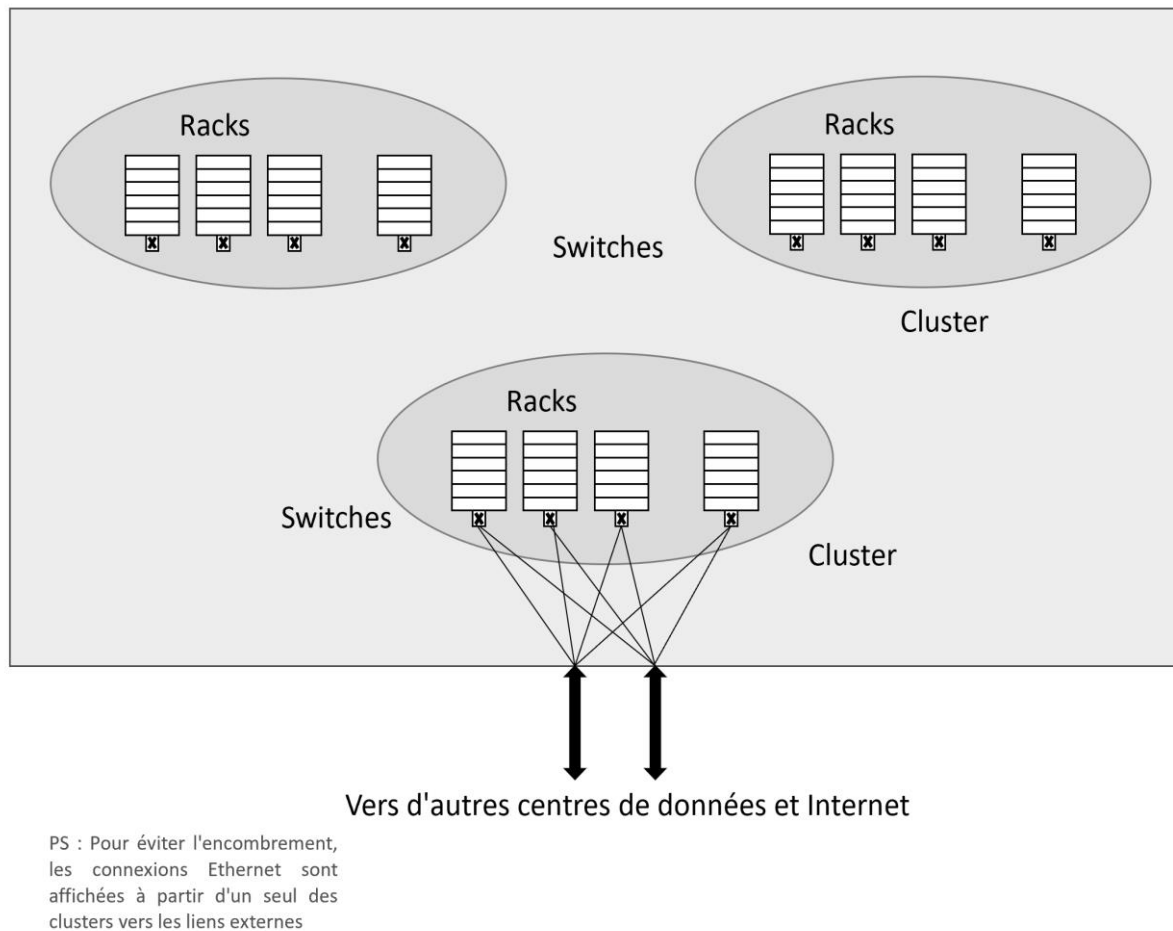


Figure 1 : Architecture de centre de données.

Nous verrons comment on utilise cette capacité de stockage et de calcul étendue et la redondance associée pour offrir des services de base.

## B. Architecture globale du système :

Avant d'examiner l'architecture globale du système, il est utile d'examiner plus en détail les défis clés :

- **Scalabilité** : la première et la plus évidente exigence pour notre infrastructure sous-jacente est de maîtriser l'évolutivité et, en particulier, d'avoir des approches qui évoluent vers ce qui est un système distribué à ultra-grande échelle. Pour notre système, on considère le problème d'évolutivité en termes de trois dimensions:
  - i) être capable de stocker plus de données.
  - ii) être capable de traiter plus de données
  - iii) être en mesure de traiter plus de requêtes

L'évolutivité nécessite l'utilisation de stratégies de systèmes distribués.

- **Fiabilité**: notre système aura des exigences de fiabilité strictes, en particulier en ce qui concerne la disponibilité des services. Ceci est particulièrement important pour la fonctionnalité de recherche, où il est nécessaire de fournir une disponibilité 24h / 24 et 7j / 7 .

Il convient de noter que l'exigence de fiabilité doit être satisfaite dans le contexte des choix de conception dans l'architecture physique, ce qui implique que les défaillances (logicielles et matérielles) sont anticipées avec une fréquence raisonnable. Cela exige à la fois de détecter les défaillances et d'adopter des stratégies pour masquer ou tolérer de telles défaillances. Ces stratégies reposent largement sur la redondance de l'architecture physique sous-jacente.

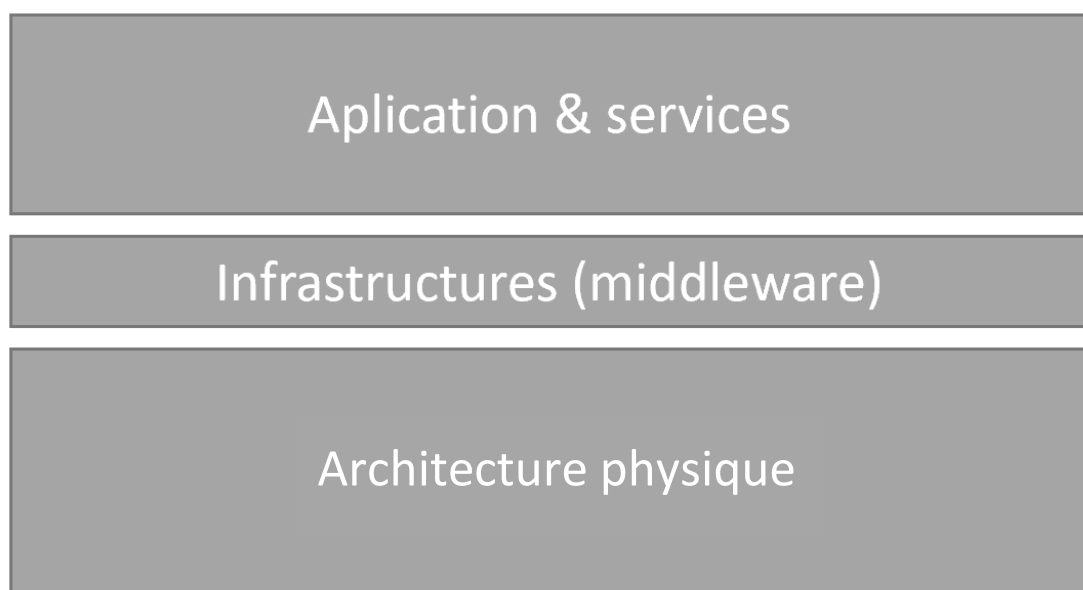


Figure 2 : L'architecture globale du système

- **Performances** : les performances globales du système sont essentielles, en particulier pour obtenir une faible latence des interactions des utilisateurs. Plus les performances sont meilleures. L'importance des performances est illustrée par l'objectif de terminer les opérations de recherche et de calcul et d'atteindre le débit requis pour répondre à toutes les demandes entrantes tout en traitant de très grands ensembles de données. Cela s'applique à un large éventail de fonctions associées au fonctionnement de notre système. Il est également important de noter que les performances sont une propriété de bout en bout qui nécessite que toutes les ressources sous-jacentes associées fonctionnent ensemble, y compris les ressources de réseau, de stockage et de calcul.
- **Ouverture** : les exigences ci-dessus sont à bien des égards les plus évidentes pour que notre système prenne en charge ses services et applications de base. Il y a également une forte exigence d'ouverture, en particulier pour soutenir le développement de la

gamme d'applications Web proposées. Ceci n'est possible qu'avec une infrastructure extensible et prend en charge le développement de nouvelles applications.

On a répondu à ces besoins en développant l'architecture globale du système illustrée à la figure suivante. Cette figure montre la plate-forme informatique sous-jacente en bas (c'est-à-dire l'architecture physique décrite ci-dessus) et les services et applications bien connus en haut.

La couche intermédiaire définit une infrastructure distribuée commune fournissant un support middleware pour la recherche et le cloud computing. Ceci est crucial pour le succès de notre système. L'infrastructure fournit les services de système distribué communs aux développeurs de services et des applications et encapsule les stratégies clés pour gérer l'évolutivité, la fiabilité et les performances. La fourniture d'une infrastructure commune bien conçue comme celle-ci peut amorcer le développement de nouvelles applications et de nouveaux services grâce à la réutilisation des services système sous-jacents et, plus subtilement, fournit une cohérence globale à la base de code croissante en appliquant des stratégies et des principes de conception communs.

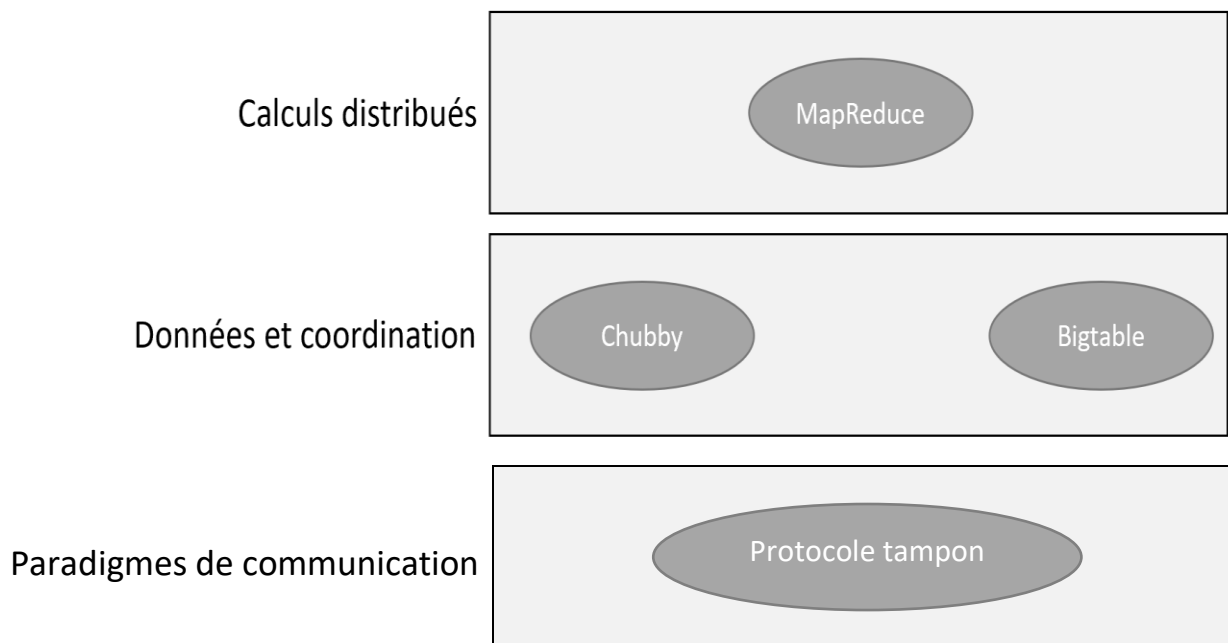


Figure 3 : Infrastructure du système.

**Infrastructure du système** : Le système est conçu comme un ensemble de services distribués offrant des fonctionnalités essentielles aux développeurs. Cet ensemble de services se divise naturellement en sous-ensembles suivants :



## 1. Les paradigmes de communication :

Les paradigmes de communication sous-jacents, y compris les services pour l'invocation à distance et la communication indirecte :

Le protocole tampons offre un format de sérialisation commun pour le système, y compris la sérialisation des demandes et des réponses lors d'une invocation à distance.

Le but d'invocation à distance est de fournir un moyen indépendant de la langue et de la plate-forme de spécifier et de sérialiser les données dans un manière simple, très efficace et extensible, les données sérialisées peuvent alors être utilisé pour le stockage ultérieur de données ou la transmission à l'aide d'un sous-jacent protocole de communication, ou même à toute autre fin qui nécessite une sérialisation format pour les données structurées.

## 2. Collection des données :

Services de données et de coordination fournissant des abstractions non structurées et semi-structurées pour le stockage des données couplées à des services pour prendre en charge un accès coordonné aux données :

- Chubby prend en charge les services de coordination et la capacité de stocker de petits volumes de données.
- Bigtable fournit une base de données distribuée offrant un accès aux données semi-structurées.

## 3. Traitement des données :

Services de calcul distribués fournissant des moyens pour effectuer des calculs parallèles et distribués sur l'infrastructure physique :

- MapReduce prend en charge le calcul distribué sur des ensembles de données potentiellement très volumineux (par exemple, stockés dans Bigtable).

MapReduce est un patron d'architecture de développement informatique, inventé par Google, dans lequel sont effectués des calculs parallèles, et souvent distribués, de données potentiellement très volumineuses, typiquement supérieures en taille à 1 téraoctet.

## 4. Visualisation des données :

La visualisation des données sera faite coté client à l'aide des applications web, qui visualisent les données à l'aide des graphes et des tableaux simplifiés.

### III. Conclusion :

En guise de conclusion, les systèmes distribués ont plusieurs raisons de leur existence.

- Partage des ressources (données, programme, services) qui permet un travail collaboratif ;
- Accès distant, c'est-à-dire qu'un même service peut être utilisé par plusieurs acteurs situés à des endroits différents ;
- Amélioration des performances : la mise en commun de plusieurs unités de calcul permet d'effectuer des calculs parallélisables en des temps plus courts ;
- Confidentialité : les données brutes ne sont pas disponibles partout au même moment, seules certaines vues sont exportées ;
- Disponibilité des données en raison de l'existence de plusieurs copies ;
- Maintien d'une vision unique de la base de données malgré la distribution ;
- Réalisation des systèmes à grande capacité d'évolution ;
- Augmentation de la fiabilité grâce à la duplication de machines ou de données, ce qui induit à une réalisation des systèmes à haute disponibilité.