Base de données

Cours 12 - Disponibilité, intégrité, et confidentialité

Table des matières

Disponibilité - Définition

La disponibilité d'une BD est un aspect les plus importants qu'un fournisseur doit pouvoir offrir **et garantir**.

La plupart d'entre eux doivent avoir un temps de disponibilité de **99.999%**.

Disponibilité - Solutions

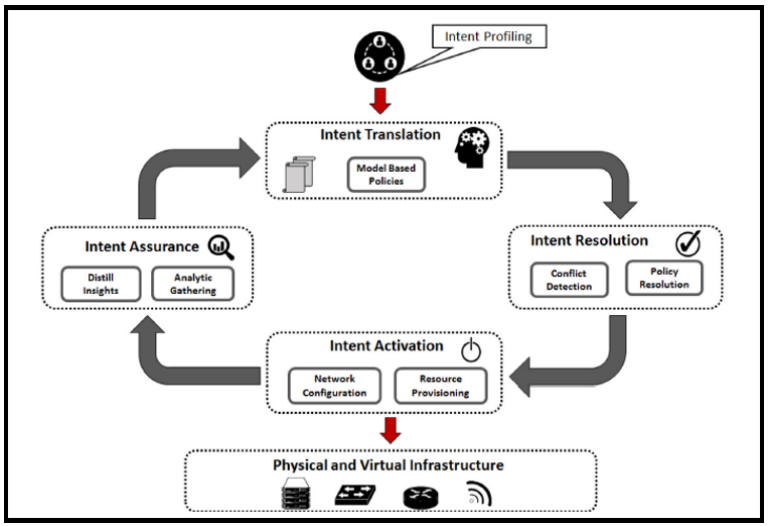


Figure: A. Leivadeas and M. Falkner, "A Survey on Intent-Based Networking," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625-655, Firstquarter 2023, doi: 10.1109/COMST.2022.3215919.

On peut améliorer la disponibilité et contrer les panes avec les solutions suivants :

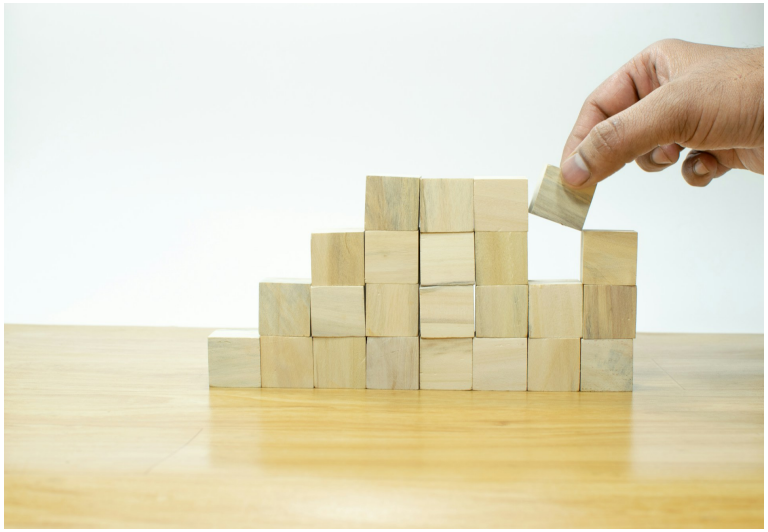
- Accès à la BD (connexions)
- Ressources (fonctionnement minimal)
- Performance (fonctionnement acceptable)
- Systèmes proactifs avec de l'IA (fonctionnement optimal)

Solutions - Accès à la BD (connexions)

Trop de connexions peuvent créer des bloquages (“bottlenecks”).

Une solution efficace est le “multi-threading” (programmation) et le “load balancing” (programmation et réseau).

Solutions - Ressources (fonctionnement minimal)



La disponibilité d'une BD est un aspect les plus importants qu'un fournisseur doit pouvoir offrir **et garantir**.

La plupart d'entre eux doivent avoir un temps de disponibilité de **99.999**%.

Utiliser des machines virtuelles (VMs) est un moyen simple de partager les ressources.

Solutions - Performance (fonctionnement acceptable)

Dans certains domaines, la disponibilité de base n'est pas suffisant pour assurer la rétention des utilisateurs.

Par exemple, il faut que les sites Web soient rapides (formulaires, chargement initial, etc.) pour ne pas perdre les utilisateurs impatients.

Ajouter du CPU, de la RAM, bande passante, etc. permet d'augmenter les performances de la BD en général.

Solutions - Systèmes proactifs avec de l'IA (fonctionnement optimal)

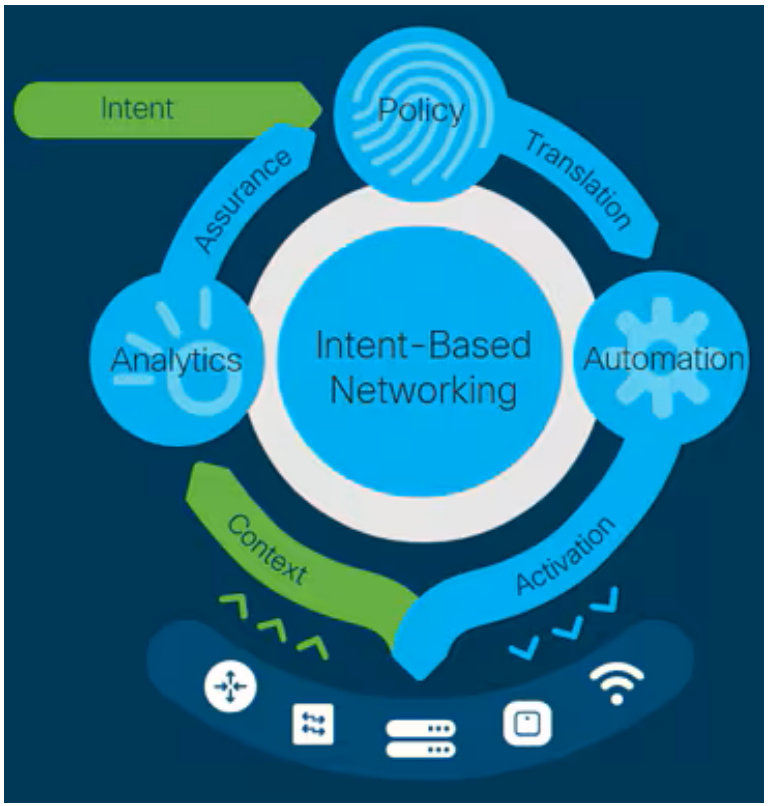


Figure: `https://www.cisco.com/c/en_sg/solutions/intent-based-networking.html`

L'Intelligence Artificielle (IA) joue un rôle important dans nos vies (ChatGPT, les voitures autonomes, etc.).

Les avancées dans ce domaine peuvent être utilisées pour faire des systèmes prédictifs comme Intent-Based Networking (IBN).

Intégrité des données - Administration d'une BD intègre

Nous avons déjà abordé l'intégrité référentielle ainsi que la cohérence au sens de la conception (structure des données).

Ceci dit, il faut en pratique faire une veille et monitorer les données.

L'administration de l'intégrité des données est un travail continu pouvant être en péril dès son contact avec le monde réel (la prod).

Intégrité des données - Techniques

Techniques principales pour assurer une administration intègre des données :

- Détection des valeurs “NULL”.
- Analyse statistique sur les données (écart type, moyenne, disparité des groupes, etc.).
- Comparaisons réactives (“TRIGGERS”).
- Une transaction (“BEGIN + COMMIT”).

Techniques - Détection des valeurs “NULL”

Les valeurs nulles/vides (“NULL”) sont une menace direct envers l’intégrité des données puisque celles-ci faussent la représentation globale (statistique) des données.

Le moyen le plus simple est de mettre la notation “NOT NULL” sur les champs.

Techniques - Analyse statistique sur les données (écart type, moyenne, disparité des groupes, etc.)

L'analyse des données est plus rattachée à l'Intelligence Artificielle (IA) et l'Intelligence d'Affaire (BI).

Pour notre cas d'application, il est très pertinent de faire un veille automatique des données en faisant un log exhaustif sur plusieurs mois (ou années) pour détecter les problématiques possibles sur la topologie des données.

Techniques - Analyse statistique sur les données (écart type, moyenne, disparité des groupes, etc.)

Voici des analyses pertinentes et fortement utilisées :

- Écart type : Mesurer la **dispersion** des données.
- Moyenne : Sur **plusieurs colonnes, isolées**, etc.
- Disparité des groupes : Voir si les groupes sont **balancés entre eux** (important pour l'intelligence artificielle).
- Min/Max : Voir la **valeur minimum et maximum** d'une colonne spécifique pour validé l'étendue des données.

Techniques - Une transaction ("BEGIN + COMMIT")

Généralement, les bases de données peuvent faire des transferts entre les différentes rangées.

Une **transaction** permet de rendre les opérations atomiques en garantissant l'exécution sans erreurs du code ou l'annulation du bloc en entier s'il y a un problème.

Techniques - Une transaction ("BEGIN + COMMIT")

Voici des exemples de transferts entre rangées :

- Banques : les transferts monétaires entre comptes.
- Jeux vidéo : la gestion des ressources (éviter les duplications).
- E-commerce : éviter l'envoi d'une commande sans avoir obtenu le paiement au préalable.
- Etc.

Techniques - Une transaction ("BEGIN + COMMIT")

Attention : la "Transaction" ne garanti pas une protection contre les erreurs de programmation vis-à-vis les requêtes.

En général, il est pertinent d'avoir les deux (Transaction + Trigger) pour être le plus sûr possible de ne pas avoir de problèmes.

Techniques - Comparaisons réactives ("TRIGGERS")

Même principe que la transaction au niveau de son utilité, le trigger programmé pour faire de la comparaison réactive va avoir pour but d'assurer que les grains et pertes lors d'un transfert entre deux rangées soient balancés.

Techniques - Comparaisons réactives ("TRIGGERS")

Ceci est d'ordre de programmation, donc il y a plusieurs manières de voir la solution :

- Garder le total précédent et le comparer au nouveau total, les deux valeurs doivent restées les mêmes.
- Assurer que le gain et perte aient les deux la différence du transfert.
- (moins rigoureux) le gain et perte ne soit pas les mêmes valeurs qu'avant l'opération.
- Etc.

Techniques - Comparaisons réactives ("TRIGGERS")

Attention : le "Trigger" ne garanti pas une protection contre les erreurs opérationnelles de la BD (un "crash" du serveur au milieu d'opération, etc.).

En général, il est pertinent d'avoir les deux (Transaction + Trigger) pour être le plus sûr possible de ne pas avoir de problèmes.

Confidentialité des données

La confidentialité est une notion fondamentale des bases de données.

Un SGBDR est complexe pour permettre d'avoir un contrôle granulé sur les permissions des acteurs interagissant avec la BD.

Granulé : terme utilisé pour spécifier un aspect très précis, spécifique, et référant à un ciblage précis.

Confidentialité des données - Techniques

Le domaine étant bien plus large, nous allons aborder seulement deux (2) principes :

- Prévenir et améliorer : Réagir après un incident (souhaitablement mineur) pour contrer les futurs incidents.
- Protéger activement : Éviter les incidents à la source avant qu'ils ne surviennent.

Prévenir et améliorer

Il est important d'avoir des processus en place pour adérer à la prévention des fuites et l'amélioration de la veille sur les données.

Il faut être le plus préventif possible contre les attaques et les manipulations non autorisées.

Il est très pertinent d'avoir ceux-ci programmés dans la base de données directement pour éviter les dépendances avec d'autres systèmes.

Prévenir et améliorer - Log des modifications dans une table

Principe : Garder une trace de chaque opération de modification depuis une table auxiliaire spécifique (une pour chaque table respective) ayant besoin d'un suivi d'accès très contrôle.

Notions nécessaires :

- CRUD sur les données
- Structure de données (création d'une table)
- "Triggers"
- Fonctions

Prévenir et améliorer - Stocker le nombre d'accès à une table avec un compteur

Principe : Avoir une table auxiliaire (au total) avec des compteurs pour toutes les tables nécessaires, et incrémenter le compteur respectif lorsqu'une opération CRUD quelconque est faite sur la table en question.

Notions nécessaires :

- CRUD sur les données
- Structure de données (création d'une table)
- "Triggers"
- Fonctions

Protéger activement

Il est **très difficile** d'assurer une sécurité à 100% vis-à-vis un dispositif (IoT, Serveur, etc.) se trouvant sur l'Internet ("World Wide Web").

Si possible, opter sur des solutions bloquant l'accès à des données confidentielles règle le problème à la source.

Ceci dit, une protection trop rigide peut rendre les tâches triviales très bureaucratiques et fastidieuses.

Protéger activement - Insertion par l'intermédiaire d'une vue

Principe : Empêcher l'accès à la vraie couche de données en offrant uniquement une vue à l'utilisateur avec des insertions simulées et transférée dans la vraie table grâce à des Triggers.

Notions nécessaires :

- CRUD sur les données
- "Triggers"
- Fonctions
- Vues ("Views")
- Usagers ("Users")

Protéger activement - Ne pas utiliser de valeurs d'affaires (NAS, etc.) comme clés

Principe : Éviter l'utilisation de colonnes (attributs) confidentiels pour optimiser la base de données.

Pourquoi ? Parce que l'index est rendu un facteur opérationel de la BD et il est difficile, en production, de changer de tels aspects en général sur les vieux systèmes.

Notions nécessaires :

- Besoin d'affaire
- Conception de BD (Schéma)

Bibliographie

- A. Leivadeas and M. Falkner, "A Survey on Intent-Based Networking," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625-655, Firstquarter 2023, doi: 10.1109/COMST.2022.3215919.
- `https://www.cisco.com/c/en_sg/solutions/intent-based-networking.html`