

Administration des bases de données

LicencePro 2006-2007
Olivier Perrin
LORIA

Organisation

2

5 séances

- ▶ séance 1: concepts
- ▶ séance 2: installation
- ▶ séance 3: configuration
- ▶ séance 4: optimisation
- ▶ séance 5: examen (deux parties)

Objectifs

- ▶ être capable d'administrer un SGBD (création d'une base, gestion, surveillance, utilisateurs, sécurité, restauration, optimisation)

Pour me contacter

- ▶ Olivier.Perrin@loria.fr

Plan

3

Motivations

Sécurité/intégrité

Sauvegarde/restauration

Performances

Gestion du changement

Mise en œuvre

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Motivations

4

SGBD

- ▶ complexe
- ▶ lié à l'organisation

Administration de qui ? de quoi ?

Administration

- ▶ des utilisateurs de la base
- ▶ des données
- ▶ des bases de données

Fonctions et rôle d'un DBA (*Database Administrator*)

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

L'utilisateur final

5

Celui qui utilise l'information

Celui qui ne sait rien (ou pas grand chose) du fonctionnement interne

Celui qui ne possède pas forcément des compétences informatiques

Celui dont l'expertise concerne le traitement des données qu'il manipule

L'information doit donc

- ▶ être correcte
- ▶ être accessible lorsque l'utilisateur en a besoin
- ▶ être en phase avec les besoins des utilisateurs
- ▶ permettre d'être productif

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Le spécialiste

6

Administrateur

Concepteur

Analyste

Programmeur

Technicien

Dépend

- ▶ de la taille de l'entreprise,
- ▶ de la complexité de l'application
- ▶ du SGBD choisi
- ▶ des utilisateurs finaux
- ▶ de l'étape du développement du projet

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Administration des données

7

Fonction managériale de haut niveau

Définition de toute l'information nécessaire d'un point de vue *gestion*

Développement et administration des règles, des procédures, des pratiques et des plans pour la définition, l'organisation, la protection et l'utilisation la plus efficace possible des données

Cela inclut toutes les données d'une organisation, informatique ou non

Différent de ...

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Administration de bases de données

8

Normalement, concerne uniquement les données traitées par informatique

Fonction de gestion et de coordination

Responsable de la partie physique de la conception:

- ▶ la sélection du logiciel et du serveur
- ▶ l'installation/mise à jour du SGBD
- ▶ la sécurité des données, intimité (*privacy*), intégrité
- ▶ la sauvegarde et la restauration
- ▶ l'amélioration des performances (requêtes, transactions, indexes)

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Administration de bases de données

9

Qu'est-ce que c'est ?

« Réussir à mettre en place et à gérer de manière satisfaisante un environnement dans lequel les données sont utilisées de manière efficace et sûre »

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Fonctions d'un DBA

10

Trois fonctions principales

- ▶ administration et gestion du contenu des données (quelles données doit-on gérer ?)
- ▶ administration et gestion de la structure des données (comment doit-on les gérer ?)
- ▶ administration et gestion des aspects physique de la base (où doit-on les gérer ?)

I. Administration et gestion du contenu des données

- ▶ quelles données seront dans la base
- ▶ quels utilisateurs et quel droits d'accès sur la base
- ▶ quels contrôles de cohérence à maintenir
- ▶ quand les données peuvent elles être supprimées

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Fonctions d'un DBA (2)

11

2. Administration et gestion de la structure des données

- ▶ combien de bases/tables/... sont nécessaires ?
- ▶ quelle est la structure des tables ?
 - normalisation (3NF, BCNF, 4NF, 5NF ?)
 - analyse fonctionnelle (quelles fonctions ?)
 - dé-normalisation (performances)
- ▶ comment implanter la sécurité ?
- ▶ comment sont effectués les contrôles de cohérence ?
- ▶ comment les accès sont-ils optimisés ?
 - transactions, cluster, indexes, requêtes
- ▶ comment et quand réorganiser la base de données ?

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Fonctions d'un DBA (3)

12

3. Administration et gestion des aspects physiques de la base

- ▶ combien de serveurs logiciels/matériels ?
 - dimensionnement des serveurs
 - performances du système d'exploitation support
- ▶ quand et comment procéder pour les sauvegardes ?
 - bases de données
 - fichiers de recouvrement
 - fichiers de logs
- ▶ procédure de démarrage/arrêt
- ▶ procédures nécessaires pour la récupération en cas de problèmes
 - export/import, réplication, rollforward/rollback

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 13

Évolution

Administrateur d'entrepôt de données (*warehouses*)
Rôle proche d'un administrateur de données ou de bases de données

Davantage orienté vers l'intégration et la coordination des données/méta-données provenant de plusieurs sources

Rôles spécifiques

- ▶ supporte des décisions
- ▶ gère la croissance de l'entrepôt
- ▶ établit des contrats sur les niveaux de service attendus

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 14

Rôle d'un DBA

Intégrité et cohérence
Surveillance et optimisation
Réduction de l'espace de stockage
Gestion du partage des données
Sécurité
Sauvegardes, restauration
Accessibilité: 7/7, 24/24
Gestion du changement

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 15

Interactions

Relation avec les utilisateurs

- ▶ besoins quant aux données
- ▶ applications prioritaires
- ▶ possession des données
- ▶ besoin en termes d'archivage
- ▶ documentation
- ▶ temps de réponse

Relation les développeurs d'applications

- ▶ contrôle de sécurité
- ▶ règles pour l'intégrité des données
- ▶ informations sur la base
- ▶ plans de tests
- ▶ formation

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 16

Interactions (2)

Relations avec l'équipe maintenance

- ▶ besoins en terme de disponibilité
- ▶ priorités sur les interventions
- ▶ procédures de sécurité
- ▶ surveillance de la performance

Relations avec développeurs de SGBD

- ▶ veille technologique
- ▶ besoins matériels
- ▶ documentation
- ▶ support en termes de services/maintenance
- ▶ utilitaires

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 17

Interactions (3)

Relations avec les vendeurs de matériel

- ▶ capacité d'un serveur/disque/...
- ▶ capacité d'extension
- ▶ incompatibilités
- ▶ capacités en termes de services/maintenance
- ▶ formation

Relations avec l'équipe de direction

- ▶ objectifs de l'entreprise
- ▶ contraintes (logiciel, matériel, temps) pour le développement, mise à jour, ...
- ▶ budget
- ▶ évolution
- ▶ changements en termes d'organisation

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 18

Règles d'or du DBA

Tout écrire !

- ▶ la documentation est plus que critique

Tout garder !

- ▶ les manuels, les programmes, les scripts, ...

Automatiser

- ▶ pourquoi faire à la main ce qui peut être automatisé ?

Partager son savoir

- ▶ plus on en sait, plus on doit partager

Analyser, simplifier, être concentré

- ▶ étant donné tout ce qui est mis en jeu...

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 19

Règles d'or du DBA (2)

Ne pas paniquer !

- ▶ il y aura des problèmes, quoique l'on fasse !

Analyser, tester, tester, tester, tester, et enfin appliquer

- ▶ clair, non ?

Comprendre le métier, pas seulement la technologie

- ▶ on prend en compte les contraintes *réelles* !

Ne pas être un ermite !

- ▶ être accessible et ouvert, malgré la réputation

Utiliser les ressources disponibles

- ▶ souvent, les problèmes ont été rencontrés par d'autres

Rester à la page !

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 20

Motivations

Sécurité/intégrité

Sauvegarde/restauration

Performances

Gestion du changement

Mise en œuvre

Olivier Perrin, LORIA LicencePro, 2006-2007

Sécurisation: définitions

Sécurité

- ▶ protection de la base de données contre les utilisateurs *non autorisés*
- ▶ vérifie que ce que les utilisateurs veulent faire est *autorisé*

Intégrité

- ▶ protection de la base de données contre les utilisateurs *autorisés*
- ▶ vérifie que ce que les utilisateurs veulent faire est *correct*

Protection des données contre les pertes accidentelles ou intentionnelles, les destructions, ou les mauvaises utilisations

Exemples

Pertes accidentelles

- ▶ erreur humaine, bug logiciel ou matériel,
- ▶ procédures pour les autorisations utilisateurs, pour les installations logicielles, pour les maintenances matérielles

Fraudes

- ▶ utilisateur, réseau, système, matériel
- ▶ contrôle des accès physiques, pare-feu, mise à jour sécurité...

Initimité/confidentialité

- ▶ données personnelles, données de l'entreprise

Intégrité des données

- ▶ données invalides/corrompues
- ▶ procédures de sauvegarde/restauration

Disponibilité

Sécurisation: différents niveaux

Au niveau du SGBD

- ▶ authentification/autorisation d'accès pour les utilisateurs de la base
- ▶ attaques sur le SGBD: failles connues, failles dans les applis associées (programmes setuid root installés par le SGBD,...)
- ▶ mauvaises configurations (mot de passe par défaut,...)

Au niveau de l'OS

- ▶ authentification/autorisation d'accès pour les utilisateurs du système
- ▶ fichiers de la base non sécurisés (lecture par tous)

Au niveau du réseau

- ▶ cryptage pour éviter interception des mots de passe, lecture des messages, lecture des fichiers de config, IP *masquerading*

Sécurisation: différents niveaux (2)

Au niveau applicatif

- ▶ utilisateurs, rôles, droits, actions possibles,...
- ▶ vues
 - restriction et vérification dynamique pour la consultation et la modification
- ▶ procédures stockées
 - contrôle des opérations et des requêtes effectuées sur la base
- ▶ injection SQL sur les applications Web

Au niveau physique

- ▶ sécurisation du stockage
- ▶ accès aux serveurs, feu, sauvegarde...

Au niveau humain

- ▶ mot de passe,...

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 25

Sécurité

Possibilités pour assurer la sécurité

- ▶ autorisations (identification, restriction) et contrôles
- ▶ vues
- ▶ procédures de cryptage pour rendre les données illisibles

Difficulté accrue avec les technologies client/serveur et le couplage Web/SGBD

- ▶ *identification*: qui
- ▶ *authentification*: l'identité est vérifiée
- ▶ *autorisation*: permission pour effectuer cette action
- ▶ *intégrité*: les données envoyées sont celles reçues
- ▶ *confidentialité*: personne n'a lu les données envoyées
- ▶ *audit*: pour vérifier a posteriori
- ▶ *non-répudiation*: preuve pour l'émetteur et le récepteur

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 26

Sécurité: autorisations

Différentes autorisations sur la base

- ▶ lecture
- ▶ insertion
- ▶ mise à jour
- ▶ suppression

Différentes autorisations sur la structure de la base

- ▶ création/suppression d'index
- ▶ ressources: création/suppression de nouvelles relations
- ▶ alteration des schémas: ajout/suppression d'attributs

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 27

Sécurité: autorisations (2)

Privilège: droit d'effectuer une action précise sur un objet donné

Actions possibles

- ▶ créer
- ▶ accéder
- ▶ modifier
- ▶ supprimer...

Notion de rôle

- ▶ possibilité de gérer un ensemble de privilèges acquis
- ▶ défini en fonction des besoins d'un certain groupe d'utilisateurs

En SQL

- ▶ `grant <liste_priv> on <relation|vue> to <liste_utilisateurs>`

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 28

Sécurité: autorisations (3)

Exemple

- ▶ une application de gestion de stock définit trois relations:
 - CLIENTS, STOCK, PRODUITS, et une vue: PRODUIT_EN_STOCK
- ▶ un utilisateur GSTOCK est défini et est le propriétaire des relations et de la vue
- ▶ un rôle de consultation est créé
 - `CREATE ROLE CONSULT_DATA`
- ▶ le privilège correspondant lui est affecté
 - `GRANT SELECT ON GSTOCK.CLIENT TO CONSULT_DATA`
- ▶ un rôle pour la mise à jour est créé
 - `CREATE ROLE UPDT_STOCK`

Olivier Perrin, LORIA LicencePro, 2006-2007

Sécurité: autorisations (4)

Suite

- ▶ les privilèges
 - GRANT SELECT, UPDATE ON GSTOCK.STOCK TO UPDT_STOCK
 - GRANT SELECT ON GSTOCK.CLIENT TO UPDT_STOCK
 - GRANT SELECT ON GSTOCK.PRODUITS TO UPDT_STOCK
 - GRANT UPDATE(NbIn, NbOut) ON GSTOCK.PRODUITS TO UPDT_STOCK
- ▶ un utilisateur WEBSTOCK de l'application Web de mise à jour du stock est créé, et le rôle UPDT_STOCK sera son rôle par défaut
 - GRANT UPDT_STOCK TO WEBSTOCK
 - ALTER USER WEBSTOCK DEFAULT ROLE UPDT_STOCK

Sécurité: règles d'autorisation

Contrôles inclus dans le système qui permettent de restreindre l'accès aux données et les actions des utilisateurs sur ces données

Une matrice des autorisations recense

- ▶ les sujets
- ▶ les objets
- ▶ les actions
- ▶ les contraintes

Chaque ligne indique que le sujet est autorisé à effectuer une action sur un objet de la base (sous certaines conditions)

Sécurité: règles d'autorisation (2)

Sujet	Objet	Action	Contrainte
DépartementVentes	Client	Insertion	LimiteCrédit<500000
DépartementCommandes	Client	Lecture	-
Anne Martin	Bon de commande	Insertion	Valeur < 2000
Programme CheckCompte	Compte	Modification	-
DépartementLivraison	Bon de commande	Suppression	-

Sécurité: tables d'autorisation

Quelques SGBDs n'implément pas ces matrices, mais utilisent des versions simplifiées:

- ▶ tables d'autorisation pour les rôles
- ▶ tables d'autorisation pour les objets

	Client	Commande	Service Ventes	Service Commandes	Comptabilité
Read	Y	Y	Y	Y	Y
Insert	Y	Y	N	Y	N
Modify	Y	N	N	Y	Y
Delete	N	N	N	N	Y

Sécurité: schémas d'authentification

Objectif: obtenir une identification de l'utilisateur

L'authentification par mot de passe n'est pas valide

- ▶ des utilisateurs peuvent partager des mots de passe
- ▶ ils sont stockés, donc ils peuvent copiés
- ▶ les connexions automatiques suppriment le besoin de rentrer son mot de passe
- ▶ les mots de passe non cryptés voyagent en clair sur le réseau

Solutions possibles

- ▶ biométrie: empreintes, rétine,...
- ▶ authentification basée sur un tiers: clés publiques/privées, certificats

Sécurité: les vues

Objectif

- ▶ assurer une indépendance logique des applications par rapport à la base (aux tables)

Moyen

- ▶ les vues: relations virtuelles dont la définition (schéma) est une requête
- ▶ elles sont interrogées et mises à jour (sous certaines conditions) comme des relations normales

Sécurité: les vues (2)

Avantages

- ▶ flexibles et efficaces
- ▶ définition de critères très proches de ce que les applications ont besoin
- ▶ définition de politiques de sécurité qui prennent en compte les données et le contexte
- ▶ invocation contrôlée sur les données
- ▶ mise à jour possible (sous conditions)

Sécurité: les vues (3)

Mettre à jour avec des vues

Conditions

- ▶ il ne manque pas d'informations (NOT NULL, clés)
- ▶ pas de distinct
- ▶ pas de fonction d'agrégation
- ▶ pas d'opérations ensembliste

Possibilités offertes

- ▶ intégrité avec CHECK OPTION
- ▶ vue possède ses propres privilèges

Sécurité: les vues (4)

37

Désavantages

- ▶ vérification des conditions d'accès peut devenir lourde
- ▶ complétude des vues par rapport à la politique de sécurité voulue
- ▶ superposition de vues (incohérences possibles)
- ▶ à compléter avec des procédures stockées

Sécurité: les contre mesures

38

Ne jamais exposer un serveur DB sur Internet

- ▶ utiliser des tunnels, un firewall avec authentification,...
- ▶ filtrer les ports du SGBD (1521,...)
- ▶ attention aux hébergeurs pas chers

Ne jamais partager un serveur BD

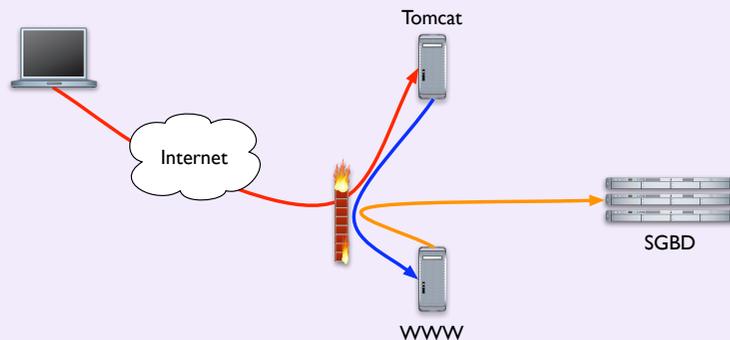
Configuration de l'OS

- ▶ permet de limiter l'exploitation des failles
- ▶ appliquer les procédures classiques
 - non installation des composants inutiles
 - limitation des services réseaux
 - application régulière des correctifs de sécurité

Sécurité: les contre mesures (2)

39

Appliquer le principe de défense en profondeur et de cloisonnement avec une architecture en strates



Sécurité: les contre mesures (3)

40

Configuration installation SGBD

- ▶ changer les mots de passe par défaut
- ▶ supprimer les comptes par défaut
- ▶ ne pas installer les exemples, les applications annexes,...

Séparation des privilèges

- ▶ recenser les rôles dans les applications (admin, maj, lecture,...)
- ▶ appliquer ces rôles dans les privilèges attribués

Audit des applicatifs

- ▶ parler de la sécurité aux développeurs
- ▶ rechercher les points critiques
- ▶ audit des sources PHP, ASP, JSP, Java, Perl,...

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 41

Intégrité

Trois catégories de contraintes

- ▶ contraintes métier (*business constraints*)
- ▶ contraintes des entités (*entity constraints*)
- ▶ contraintes référentielles (*referential constraints*)

Contrainte métier

- ▶ la valeur d'un attribut est contrainte par la valeur d'un autre attribut ou par le résultat d'une formule

Contrainte d'entité

- ▶ un attribut d'une relation est contraint, par exemple, NOT NULL

Contrainte référentielle

- ▶ contrainte liée aux clés (la relation B dépend de la relation A)

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 42

Intégrité

Les avantages offerts par les contraintes

- ▶ garantie pour l'intégrité et la cohérence
- ▶ définies comme partie intégrante de la table
- ▶ s'applique à toutes les applications utilisant la table
- ▶ ne peuvent pas être détournées
- ▶ augmente la productivité du développement
- ▶ pas de programmation nécessaire
- ▶ facile à spécifier et à maintenir
- ▶ définies en une fois à un seul endroit

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 43

Intégrité: techniques

Contrôles d'intégrité

- ▶ protéger les données d'un usage non autorisé
- ▶ contraintes lors de la création
- ▶ domaines
- ▶ triggers
- ▶ vues

Raffinement des contraintes dans les applications

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 44

Intégrité: contraintes à la création

Clé primaire

Clé étrangère et référence

Null/not null

Check

```
create table Compte(
  compteId char(6) primary key,
  compteClientId char(6) references Client(clientId),
  compteOrdre number(4) check (compteOrdre <= 200),
  compteType char(2) constraint typeChk
    check (compteType in ('AB','CD','EF')),
  compteDate date not null
);
```

Olivier Perrin, LORIA LicencePro, 2006-2007

Intégrité: domaine

45

Domaine

- ▶ création d'un type de données par l'utilisateur et des valeurs autorisées

```
CREATE DOMAIN DomainePrixAug AS DECIMAL
CHECK(VALUE BETWEEN .001 and 0.15);
```

- ▶ dans une relation, on peut alors avoir

```
AugmentationPrix DomainePrixAug NOT NULL
```

- ▶ avantage par rapport à un CHECK: un seul point d'entrée pour une modification ultérieure

Intégrité: assertions

46

Assertions

- ▶ contraintes permettant de renforcer les conditions sur la base
- ▶ elles sont automatiquement vérifiées lorsque qu'une instruction agit sur une table ou un attribut sur lequel existent des assertions
- ▶ exemple: aucun employé ne peut diriger son épouse ou son époux

```
CREATE ASSERTION Direction
CHECK (DirecteurID <> EpouseID)
```

Intégrité: triggers

47

Triggers

- ▶ basés sur les règles ECA, *Event - Condition - Action*
- ▶ l'intérêt des triggers concerne le fait que le contrôle est effectué
 - pour tous les utilisateurs
 - pour toutes les activités
- ▶ le contrôle n'a pas à être codé pour chaque requête ou programme
- ▶ un programme ne peut pas détourner la règle établie par un trigger
- ▶ inconvénient: difficile de savoir ce qui se passe

Intégrité: vues

48

Vues et sous-schémas permettant de restreindre les vues de l'utilisateur sur la base

Vues

- ▶ sous-ensemble de la base présenté à un ou plusieurs utilisateurs
- ▶ accorder des privilèges d'accès à la vue plutôt qu'à la table

Option CHECK OPTION

Motivations
Sécurité/intégrité
Sauvegarde/restauration
Performances
Gestion du changement
Mise en œuvre

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Sauvegarde

Copie (*dump*) de la base qui permet d'avoir une image (partielle ou totale) de la base à un instant donné

Périodicité

- ▶ journalière
- ▶ hebdomadaire
- ▶ mensuelle

Type

- ▶ *cold backup*: la base est arrêtée pendant la sauvegarde
- ▶ *hot backup*: un fragment est arrêté et sauvegardé

Les copies sont entreposées dans un site sûr !

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Sauvegarde: méthodologie

À chaud (on-line)

- ▶ minimise la durée de la sauvegarde
- ▶ peut provoquer des blocages (en fonction du SGBD)
- ▶ nécessite la mise en œuvre de méthodologie spécifique en fonction du SGBD

À froid (off-line)

- ▶ minimise la durée de la restauration
- ▶ nécessite un arrêt de la base
- ▶ effectuée par copie de l'ensemble des éléments de la base
- ▶ compatible avec tous les SGBD

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre

Sauvegarde: stratégie

Souvent une combinaison de sauvegarde à chaud et à froid

- ▶ à froid une fois tous les N jours
- ▶ à chaud plus régulièrement

On joue sur le mode et la durée de rétention

- ▶ redondance: combien de copies à un temps t donné
- ▶ durée: pendant combien de temps une sauvegarde doit-elle être conservée ?

Olivier Perrin, LORIA LicencePro, 2006-2007

Sauvegarde: stratégie (2)

Nombre de bandes nécessaires ?

- ▶ méthode GFS: grand-père, père, fils
 - fils: une sauvegarde différentielle ou FULL journalière
 - père: une sauvegarde FULL par semaine
 - grand-père: une sauvegarde FULL par mois
- ▶ rotation des bandes sur un an
 - un jeu de bandes est utilisé pour le fils (4, 5, ou 6)
 - un jeu de bandes est utilisé pour le père (4)
 - un jeu de bandes est utilisé pour le grand-père (12)

Restauration de la base

Mécanisme permettant de récupérer rapidement et dans un état correct une base après un problème logiciel ou matériel

- ▶ crash système
- ▶ problèmes des éléments de stockage
- ▶ erreurs au niveau des applications
- ▶ problème physiques (inondations, feu, ...)
- ▶ sabotage

Possibilités

- ▶ sauvegarde (*backup*)
- ▶ journalisation (*journalizing*)
- ▶ point de contrôle (*checkpoint*)
- ▶ gestion de sauvegarde (*recovery manager*)

Transactions

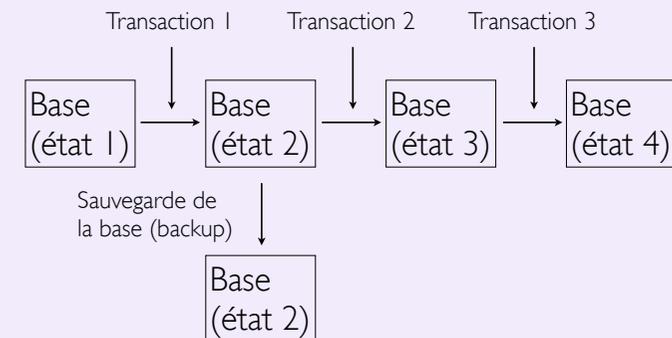
Unité de base pour le recouvrement

Propriétés d'une transaction

- ▶ Atomicité
- ▶ Cohérence
- ▶ Isolation
- ▶ Durabilité

Le rôle du gestionnaire de recouvrement est d'assurer l'atomicité et la durabilité

Mise à jour d'une base



Journalisation

Vérification rétrospective des transactions et des mises à jour de la base de données

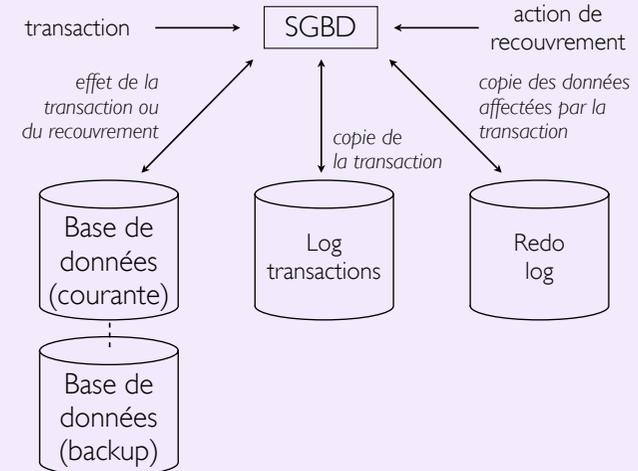
Log des transactions

- ▶ enregistrement des données pour chaque transaction exécutée

Log des modifications de la base

- ▶ images des données modifiées
- ▶ *before-image*: copie avant modification
- ▶ *after-image*: copie après modification

Journalisation (2)



Journalisation (3)

TId	Time	Operation	Object	Before	After	pPtr	nPtr
T1	10:12	Start				0	2
T2	10:13	Update	TENANT NO21	(old value)	(new value)	1	8
T2	10:14	Start				0	4
T2	10:16	Insert	TENANT NO37		(new value)	3	5
T2	10:17	Delete	TENANT NO9	(old value)		4	6
T2	10:17	Update	PROPERTY PG16	(old value)	(new value)	5	9
T1	10:18	Commit				2	0
	10:19	Checkpoint	T2				

pointeurs vers les logs précédent et suivant pour chaque transaction

Journalisation (4)

Redo logs - *after images*

- ▶ après que tout attribut d'un n-uplet d'une table ait été modifiée, les nouvelles valeurs sont écrites dans la base et dans le fichier Redo Log
- ▶ le n-uplet est écrit dans le Log
- ▶ si le n-uplet est supprimé, cela est écrit dans le Redo Log

Redo logs - *before images*

- ▶ avant qu'un n-uplet soit modifié, les valeurs sont copiées dans le Redo Log
- ▶ il ne s'agit pas d'une simple copie provenant de la base, mais d'une copie venant du ROLLBACK SEGMENT qui stocke les valeurs des données avant leur écriture

Point de contrôle

La base et les fichiers de logs des transactions sont synchronisés

- ▶ le système est en mode *quiet*
- ▶ le SGBD refuse d'accepter de nouvelles transactions pendant ce temps

Cela permet au gestionnaire de recouvrement d'agir sur des périodes courtes plutôt que de bloquer la base sur une longue période

Procédures de recouvrement et reprise

Réplication: bases en miroir

Restauration/ré-exécution: relancer les transactions sur la copie depuis le dernier état correct

Intégrité des transactions: validation ou annulation de toutes les transactions

Recouvrement en arrière (*backward recovery*): application des *before images*

Recouvrement en avant (*forward recovery*): application des *after images*

- ▶ préférable à restauration/ré-exécution

Réplication

Nécessite deux copies de la base

Avantages

- ▶ recouvrement rapide (secondes)
- ▶ adapté aux problèmes liés aux disques

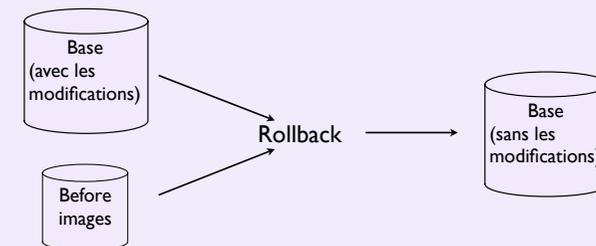
Inconvénients

- ▶ pas de protection contre les pannes électriques
- ▶ coûteux

Recouvrement en arrière

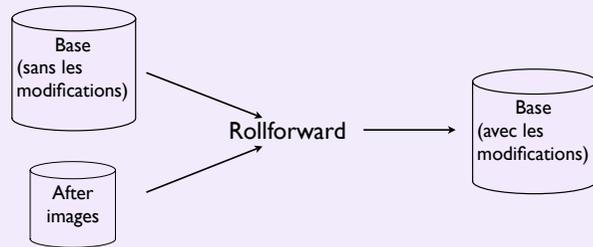
Les modifications faites sur la base sont défaites

Cela permet de revenir à un état de la base qui soit cohérent



Recouvrement en avant

On part d'un état correct mais ancien de la base, et on applique les effets des transactions dans l'ordre



Ré-exécution des transactions

Proche du recouvrement en avant

Utilise les transactions plutôt que l'image des effets

Avantages

- ▶ simple

Inconvénient

- ▶ lent et long

Réponse aux échecs

Transactions abandonnées

- ▶ mécanisme privilégié: rollback
- ▶ alternative: rollforward à l'état juste avant l'annulation

Données incohérentes

- ▶ mécanisme privilégié: rollback
- ▶ alternatives: ré-exécuter les transactions qui n'engendrent pas d'incohérences ou transactions de compensation

Problème système

- ▶ mécanisme privilégié: migration vers la base sauvegardée
- ▶ alternatives: rollback ou redémarrage du point de contrôle

Destruction de la base

- ▶ mécanisme privilégié: migration vers la base sauvegardée
- ▶ alternatives: rollforward ou ré-exécution des transactions

Motivations

Sécurité/intégrité

Sauvegarde/restauration

Performances

Gestion du changement

Mise en œuvre

Performances ?

- De quelles performances parle-t-on ?
- Quels sont les éléments actifs et quels sont leurs rôles respectifs ?
- Quelles sont les mesures de performance significatives sur ces éléments ?
- Quelles sont les optimisations possibles ?
- Comment les mettre en œuvre ?

Critères de performances

- Notion de performance: capacité à répondre dans les délais prévus aux besoins prévus dans le contexte prévu
- Donc, la question essentielle est:
 - « Qu'est-ce qui est prévu ? »

Critères de performances (2)

- Questions:
- ▶ combien d'utilisateurs sont prévus
 - ▶ comment les utilisateurs vont-ils interagir ?
 - ▶ quelle est la vitesse du réseau ?
 - ▶ quelle est la quantité de données seront accédées par les utilisateurs ?
 - ▶ quelle proportion en lecture seule ?
 - ▶ quel est le temps de réponse exigé par les utilisateurs ?
 - ▶ le système doit-il être 24/24 ?
 - ▶ les changements doivent-ils être pris en compte en temps réel ?

Où agir ?

- Côté infrastructure sous-jacente
 - ▶ système d'exploitation
 - ▶ ressources matérielles
- Côté instance de la base
 - ▶ réglage des paramètres mémoire et buffers
 - ▶ élimination des blocages
 - ▶ surveillance des E/S sur les données
- Côté applicatif
 - ▶ conception de la base de données
 - ▶ codage de l'application (source de 80% des pbs de performance)

Système sous-jacent

Au niveau de l'installation

- utilisation appropriée des paramètres d'installation

Au niveau de la mémoire

- utilisation optimale de la mémoire
- interfaces des disques (SCSI, IDE, ...)
- adaptation des caches

Au niveau du réseau

- charge
- débit

Système sous-jacent (2)

Au niveau des entrées/sorties

- placement des données (données adjacentes, répartition)
- réplication (*mirroring*) des données les plus accédées

Au niveau du CPU

- surveillance de la charge
- calcul des priorités des processus

SGBD/Instance de la base

Au niveau SGBD: réglage des paramètres mémoire et buffers

- caches
- logs
- mémoire allouée aux tris
- mémoire allouée aux rollback
- ...

Au niveau de l'instance

- utiliser les informations fournies par les statistiques
- concurrence

Application

Au niveau conception de la base

- normalisation/dénormalisation
- vues
- index

Au niveau codage

- optimisation des programmes SQL pour réduire le temps de réponse et/ou les ressources nécessaires
- moyens
 - réduire la charge
 - organiser la charge
 - paralléliser la charge

Performances

77

Optimisation (*tuning*) à plusieurs niveaux

- ▶ installation
 - utilisation appropriée des paramètres d'installation
- ▶ mémoire
 - utilisation optimale de la mémoire
 - adaptation des caches
 - calcul des priorités de processus
- ▶ entrées/sorties
 - placement des données (données adjacentes, répartition - *striping*)
 - réplication (*mirroring*) des données les plus accédées
- ▶ CPU
 - surveillance de la charge
- ▶ code
 - optimisation des programmes SQL

Concurrence

78

Problèmes liés à la concurrence

Dans un environnement multi-utilisateurs

- ▶ accès simultanés aux données
- ▶ interférences
- ▶ pertes de données possibles

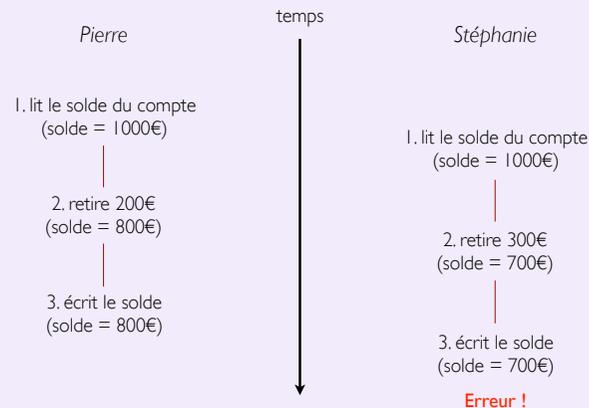
Solution: contrôle de la concurrence

- ▶ pas d'interférence entre les opérations
- ▶ l'intégrité des données est vérifiée

Concurrence (2)

79

Exemple: perte de mise à jour



Concurrence (3)

80

Techniques de contrôle de la concurrence

Sérialisabilité

- ▶ terminer une transaction avant d'en commencer une autre

Mécanismes de verrouillage

- ▶ la façon la plus commune de réaliser la sérialisabilité
- ▶ les données qui sont censées être modifiées sont verrouillées pour l'utilisateur qui va les mettre à jour
- ▶ aucun autre utilisateur ne peut effectuer de mise à jour tant que le verrou existe

Concurrence (4)

81



Verrouillage

82

Mécanismes de verrouillage

Niveau de verrouillage

- ▶ base - pendant les mises à jour de la base
- ▶ table - pour les mises à jour en rafale
- ▶ bloc ou page - le plus utilisé
- ▶ enregistrement - seulement les lignes demandées
- ▶ champ - nécessite beaucoup de travail annexe, peu pratique

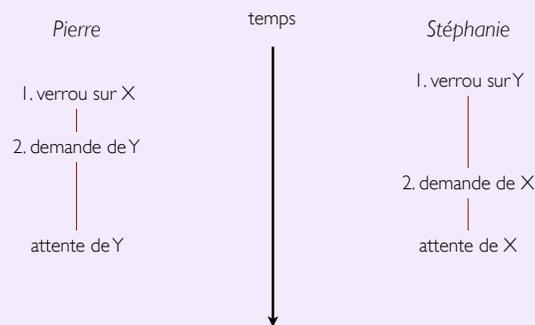
Type de verrou

- ▶ partagé - lecture, mais pas de mise à jour possible. Utilisé pour lire et pour éviter que d'autres posent un verrou exclusif
- ▶ exclusif - pas d'accès permis. Utilisé pour préparer une mise à jour

Verrouillage (2)

83

Deadlock: impasse quand deux ou plusieurs transactions ont verrouillé une ressource commune, et chacun attend que l'autre libère le verrou



Verrouillage (3)

84

Gestion des interblocages (deadlocks)

Prévention

- ▶ opérer toutes les opérations de verrouillage avant la première opération de déverrouillage
- ▶ protocole *2-phase locking* (verrouillage à 2 phases)
 - phase d'expansion: tous les verrous sont apposés
 - phase de rétrécissement: les verrous sont libérés et aucun nouveau verrou ne peut être apposé

Résolution

- ▶ autoriser l'apparition d'interblocages
- ▶ mécanisme pour les détecter et les casser
 - matrice d'utilisation des ressources

Verrouillage (4)

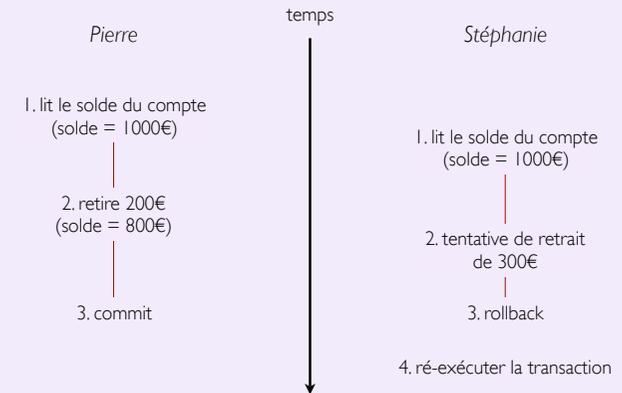
85

Versionnement

- ▶ approche optimiste du contrôle de la concurrence
- ▶ remplace le verrouillage
- ▶ hypothèse: les mises à jour concurrentes sont peu fréquentes
- ▶ une transaction peut tenter une mise à jour n'importe quand
- ▶ le système rejette la mise à jour s'il pressent un conflit
- ▶ utilisation de rollback et commit

Verrouillage (5)

86



87

Motivations

Sécurité/intégrité

Sauvegarde/restauration

Performances

Gestion du changement

Mise en œuvre

Gestion du changement

88

En entreprise, le changement peut être:

- ▶ déménagement physique
- ▶ modification organisationnelle et/ou des procédés
- ▶ modification de l'infrastructure réseau
- ▶ évolution des applications avec des données existantes
- ▶ modification des types et structures de données avec des données existantes

Modification des applications

- ▶ changement des spécifications
- ▶ évolution des performances/nécessité d'optimisation
- ▶ augmentation du volume de données
- ▶ prise en compte de nouvelles technologies

Gestion du changement (2)

- Anticipation: le plus tôt est le moins coûteux
- Réflexion: réfléchir à l'ensemble des aspects mis en jeu
- Planification: maximiser l'efficacité
- Information: mettre au courant les utilisateurs
- Analyse d'impact: analyser les liens entre les éléments
- Automatisation: réduire les risques d'erreurs humaines
- Continuité de service: minimiser le temps nécessaire à la procédure de changement
- Livraison rapide et efficace: le contraire est dramatique !

Gestion du changement (3)

- Le DBA devient maître d'œuvre
- Il est rarement à l'origine des modifications
 - il connaît la rigidité des SGBD
 - il connaît la complexité de l'infrastructure
 - il connaît la complexité du processus de changement
- C'est à lui que revient la charge de rendre le changement le plus transparent possible !

Types de changements: upgrade du SGBD

Avantages

- nouvelles fonctionnalités dont les développeurs ont besoin
- ajustement des pré-requis de systèmes propriétaires
- meilleures performances et stabilité
- support du constructeur
- remise au même niveau des systèmes de développement et production

Inconvénients

- interruption de l'application
- disparition de fonctionnalités
- coût
- réécriture du code
- support moins performant

Types de changements: autour du SGBD

- Modification du matériel
- Modification/cohérence de l'analyse/conception
 - MCD/MCT
 - MLD/MLT
- Modification de l'application
- Modification des technologies

Types de changements: modification de la base

Ce qu'ALTER ne peut pas faire...

Base

- ▶ changer son nom
- ▶ déplacer un objet d'une base à l'autre

Structure physique

- ▶ changer le nombre de partitions ou de fichiers de données

Tables

- ▶ modifier l'ordre des attributs
- ▶ effacer un attribut, en changer le type, la taille
- ▶ ajouter un attribut entre deux autres

Types de changements: modification de la base

Vues

- ▶ ajouter ou effacer une colonne, changer le SELECT

Index

- ▶ unicité, clustering, ordre

Modifier un trigger

Exemple: ajout d'un attribut entre deux autres ?

Types de changements: modification de la base

- ▶ retrouver la définition de la table
- ▶ retrouver la définition de toute vue qui la référence
- ▶ retrouver la définition de tout index qui la référence
- ▶ retrouver la définition de tout trigger sur la table
- ▶ examiner toutes les contraintes sur la table et mesurer l'impact de l'effacement de la table
- ▶ retrouver tous les grants sur la table
- ▶ retrouver tous les programmes qui la référence
- ▶ sauvegarder les données de la table
- ▶ effacer la table (cela inclut index, triggers, vues)
- ▶ recréer la table avec la nouvelle colonne
- ▶ recharger les données
- ▶ recréer les contraintes, les triggers, les index, les vues, les grant
- ▶ examiner les requêtes applicatives qui référencent la table

En cas de problème...

Disfonctionnement d'un média physique

- ▶ disque qui rend l'âme
- ▶ CPU qui rend l'âme

Erreur de manipulation

- ▶ utilisateur qui effectue une opération « malheureuse »

Corruption de l'état de l'instance

- ▶ suite à un arrêt brutal

Erreur de paramétrage

Problème réseau

En cas de problème... (2)

Le mieux: avoir une solution préventive !

Matériel

- sauvegarde, cluster

Manipulation

- sauvegarde, instances de développement

Corruption

- sauvegarde, gestion de l'instance par le DBA

Paramétrage

- actions ponctuelles

Réseau

- surveillance du réseau

Conclusion

Gestion de la qualité des données

Lien avec tous les utilisateurs

5 qualités

- sécurité et récupération
- contrôle des utilisateurs
- contrôle des accès
- maintenance (logicielle & matérielle)
- protection des données

Plan

Motivations

Sécurité/intégrité

Sauvegarde/restauration

Performances

Gestion du changement

Mise en œuvre

Quel SGBD ?

Open source

- MySQL 5.x
- PostgreSQL 8.x
- Firebird 1.5.x
- MaxDB 7.5
- Ingres R3

Open source ≠ gratuit: bien analyser les licences

Bien analyser les besoins et les fonctionnalités

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 101

Analyse

Quelques critères

- licence
- plates-formes
- compatibilité SQL standard
- vitesse
- stabilité
- compatibilité ACID
- intégrité
- sécurité
- authentification
- vues
- ...

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 102

Analyse (2)

Licence

- MySQL
 - GPL si vous fournissez tout le source de votre application
 - Commerciale sinon
- PostgreSQL
 - BSD licence (pas besoin d'inclure le source)
- Firebird
 - IDPL & IPL (Interbase Public Licence)

Commentaire

- attention à la licence !

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 103

Analyse (3)

Plateformes

- MySQL
 - Linux, MacOS, Windows, FreeBSD, OpenBSD, ...
- PostgreSQL
 - Linux, MacOS, Windows, FreeBSD, OpenBSD, ...
- Firebird
 - Linux, FreeBSD, MacOS, Windows

Commentaires

- toutes les plateformes les plus courantes sont supportées

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 104

Analyse (4)

Conformité SQL

- MySQL
 - moyenne
- PostgreSQL
 - très bonne
- Firebird
 - bonne

Commentaires

- MySQL: sous-ensembles de SQL 92 et SQL 99
- PostgreSQL: sous-ensemble de SQL 2003

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 105

Analyse (5)

Vitesse

- ▶ MySQL
 - moyenne/bonne
- ▶ PostgreSQL
 - moyenne
- ▶ Firebird
 - moyenne/bonne

Commentaires

- ▶ MySQL utilise un serveur avec des threads, PostgreSQL utilise les processus
- ▶ MySQL supérieur pour SELECT simple, moins pour UPDATE
- ▶ PostgreSQL largement optimisable

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 106

Analyse (6)

Stabilité

- ▶ MySQL
 - bonne
- ▶ PostgreSQL
 - bonne
- ▶ Firebird
 - bonne

Commentaire

- ▶ toujours prendre une version stable !

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 107

Analyse (7)

Sécurité

- ▶ MySQL
 - bonne
- ▶ PostgreSQL
 - bonne
- ▶ Firebird
 - ?

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 108

Analyse (8)

Stockage

- ▶ MySQL
 - plusieurs possibilités: MyISAM, BerkeleyBD, MERGE, InnoDB
- ▶ PostgreSQL
 - une seule possibilité
- ▶ Firebird
 - EVT

Commentaire

- ▶ MySQL: seul InnoDB est *transaction-safe*
- ▶ PostgreSQL: un seul modèle, plus cohérent

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 109

Analyse (9)

Intégrité des données

- ▶ MySQL
 - ACID classique
- ▶ PostgreSQL
 - MVCC: Multi Version Concurrency Control
- ▶ Firebird
 - ACID

Commentaire

- ▶ MySQL: row-level
- ▶ PostgreSQL: row-level, MVCC

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 110

Analyse (10)

Procédures stockées

- ▶ MySQL
 - oui, supérieure à 5.0
- ▶ PostgreSQL
 - oui
- ▶ Firebird
 - oui

Commentaire

- ▶ PostgreSQL: langage pl/pgSQL, plus pl/Perl, pl/TCL, pl/Python

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 111

Analyse (11)

Langage

- ▶ MySQL
 - non
- ▶ PostgreSQL
 - oui
- ▶ Firebird
 - non

Commentaire

- ▶ PostgreSQL pgSQL est équivalent à PL/SQL (Oracle)

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 112

Analyse (12)

Triggers

- ▶ MySQL
 - oui, rudimentaire, version supérieure à 5.0
- ▶ PostgreSQL
 - oui
- ▶ Firebird
 - oui

Commentaire

- ▶ pas de contraintes pour MySQL

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 113

Analyse (13)

Index

- ▶ MySQL
 - oui, full-text par tierce partie
- ▶ PostgreSQL
 - oui, full-text en modifiant le source
- ▶ Firebird
 - oui

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 114

Analyse (14)

Backups

- ▶ MySQL
 - InnoDB seulement pour la cohérence
 - réplication
- ▶ PostgreSQL
 - Write Ahead Logging
 - Recovery après un problème disque (Point-in-Time Recovery)
- ▶ Firebird
 - oui, backup incrémental

Commentaire

- ▶ réplication single-master ou multi-slave pour MySQL et PostgreSQL

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 115

Analyse (15)

Interface

- ▶ MySQL
 - JDBC, ODBC, C++, Python, Perl, PHP, ADO.Net
- ▶ PostgreSQL
 - ODBC, JDBC, C++, Python, Perl, PHP, Tcl/Tk
- ▶ Firebird
 - ODBC, JDBC, PHP, Perl, Python, ADO.Net

Commentaire

- ▶ PostgreSQL possède en plus son propre langage

Olivier Perrin, LORIA LicencePro, 2006-2007

Motivations Sécurité Sauvegarde Performance Changement Mise en œuvre 116

Choix de ...

PostgreSQL version 8

Olivier Perrin, LORIA LicencePro, 2006-2007

Mise en œuvre

Télécharger

- ▶ <http://www.postgresql.org/ftp/source/v8.1.4/postgresql-8.1.4.tar.gz>

Compiler

- ▶ outils
 - gmake, gcc, gzip, tar
- ▶ packages
 - GNU Readline library
 - OpenSSL
 - Tcl/Tk
 - JDK (JDBC)
- ▶ espace disque
 - 200 Mo nécessaires pour installation
 - l'espace évolue en fonction des données

Mise en œuvre (2)

Construire l'application

```
./configure
gmake
su
gmake install
adduser postgres
mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start
/usr/local/pgsql/bin/createdb test
/usr/local/pgsql/bin/psql test
```

Mise en œuvre (3)

Options dans configure

- ▶ --enable-multibyte: pour les caractères
- ▶ --with-maxbackends=NUMBER (32 connexions par défaut)
- ▶ --with-perl, --with-python, --with-tcl: interfaces
- ▶ --enable-odbc: support de ODBC
- ▶ --with-openssl=DIRECTORY: connexions SSL
- ▶ --with-java: support de JDBC
- ▶ --enable-syslog: démon syslog
- ▶ --enable-debug: symboles de debug
- ▶ --enable-cassert: assertions (dégrade les performances)

Environnement

L'utilisateur PostgreSQL

- ▶ le serveur est démarré à partir d'un compte utilisateur spécial
- ▶ généralement, **postgres**
- ▶ ajouté grâce à **adduser**

Ne pas utiliser le compte ROOT !

Espace disque

121

Créer un cluster

- ▶ première chose, initialisation d'un espace sur le disque
- ▶ notion de cluster
 - collection de bases de données
 - gérées par une seule instance du serveur
- ▶ après initialisation, le cluster contient une base **template**
- ▶ le cluster est créé sous `/usr/local/pgsql/data` grâce à **initdb**
- ▶ `initdb -D /usr/local/pgsql/data` à partir du compte **postgres**

Espace disque (2)

122

Créer un cluster (suite)

- ▶ options
 - `-W`: nécessite un mot de passe administrateur pour créer le cluster
 - `-A md5, -A password`: le mode d'authentification trust n'est pas utilisé
 - `-D directory`: répertoire des données
 - `-E encoding`: caractères
 - `-U username`: nom de l'administrateur
 - `-d`: debug

Démarrer

123

Démarrer le serveur

- ▶ utilisation de `pg_ctl`
 - `pg_ctl start`
 - `pg_ctl stop`
 - `pg_ctl restart`
 - `pg_ctl status`
 - `pg_ctl reload`
 - `pg_ctl kill`
- ▶ options
 - `-D directory`: répertoire des données
 - `-l logfile`: fichier de log
 - `-m mode`: mode de l'arrêt: smart, fast, immediate (smart est conseillé !)
 - `-w/-W`: attend ou n'attend pas que le lancement soit fini pour redonner la main
 - `-o options`: options de `postmaster`

Démarrer (2)

124

Démarrage (suite)

- ▶ le serveur doit savoir où trouver les données
- ▶ `pg_ctl -D /usr/local/pgsql/data -l logfile start`

```
[galibier@test ~]$ pg_ctl -D /usr/local/pgsql/data start
postmaster successfully started
DEBUG: database system was shut down at 2006-09-18 08:06:34 PDT
DEBUG: CheckPoint record at (0, 1000524052)
DEBUG: Redo record at (0, 1000524052); Undo record at (0, 0);
Shutdown TRUE
DEBUG: NextTransactionId: 815832; NextOid: 3628113
DEBUG: database system is in production state
```

Arrêter

Ne jamais faire de kill -KILL (ou kill -9)

- perte de données ou données corrompues

Arrêter avec pg_ctl

- smart: attend que tous les clients aient fermé leur connexions
- fast: le serveur s'arrête, sans vérifier les clients
- immediate: contourne la procédure d'arrêt, recovery au redémarrage

```
[galibier@test ~]$ pg_ctl -D /usr/local/pgsql/data stop -m fast
Fast Shutdown request at Mon Sep 18 09:23:39 2006
DEBUG: shutting down
waiting for postmaster to shut down.....
DEBUG: database system is shut down
done
postmaster successfully shut down
```

Redémarrage & statut

Option restart de pg_ctl

- équivalent à stop et start

Option status

- pas d'impact sur les données
- nécessite -D chemin

Option reload

- relit les fichiers de configuration: pg_hba.conf, postgresql.conf, ...
- envoi d'un signal SIGHUP (relecture)

Option kill

- pour Windows