

Plan

Introduction

Architectures
transactionnellesApproche à base de
composantsServeurs
d'applications

NFP111 — Systèmes et Applications Réparties

Cours 10 - Les Enterprise Java Beans (Introduction aux serveurs d'applications)

Claude Duvallet

Université du Havre
UFR Sciences et Techniques
25 rue Philippe Lebon - BP 540
76058 LE HAVRE CEDEX

Courriel : Claude.Duvallet@gmail.com
<http://litis.univ-lehavre.fr/~duvallet/>

Plan

Introduction

Architectures
transactionnelles

Approche à base de
composants

Serveurs
d'applications

Introduction aux serveurs d'applications

1 Introduction

2 Architectures transactionnelles

3 Approche à base de composants

4 Serveurs d'applications

Plan

Introduction

Architectures
transactionnelles

Approche à base de
composants

Serveurs
d'applications

Introduction aux serveurs d'applications

1 Introduction

2 Architectures transactionnelles

3 Approche à base de composants

4 Serveurs d'applications

Plan

Introduction

Architectures
transactionnelles

Approche à base de
composants

Serveurs
d'applications

Introduction aux serveurs d'applications

- 1 Introduction
- 2 Architectures transactionnelles
- 3 Approche à base de composants
- 4 Serveurs d'applications

Plan

Introduction

Architectures
transactionnellesApproche à base de
composantsServeurs
d'applications

Contexte

- Vers une évolution des architectures informatiques :
 - Le modèle client-serveur.
 - Les systèmes à objets distribués.
⇒ de nombreuses limitations.
- Des architectures multi-niveaux : les serveurs d'applications
 - Assurer la logique métier des applications en découplant celle-ci des aspects présentation et accès aux données.
- Fonctionnalités des serveurs d'applications :
 - gestion transactionnelle,
 - administration et exploitation,
 - sécurité et robustesse,
 - utilisation de composants COM et EJB.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Les architectures transactionnelles

- Issu des technologies “mainframe” d’IBM des années 70.
- Le terme « transaction » désigne le fait que plusieurs écrans peuvent s’enchaîner avant qu’une modification ne soit effective.
- Repris dans le monde des SGBD.
- Le modèle Distributed Transaction Processing (DTP) :
 - proposé par l’X/Open, il offre deux interfaces :
 - TX : interface entre le Transaction Manager (TM) et l’applicatif. Une API qui permet d’invoquer le moniteur transactionnel.
 - XA : interface du Ressource Manager (RM) directement utilisée par le moniteur transactionnel.
 - Mise en œuvre de protocoles de validation à deux phases (Two-Phase Commit) : préparation de toutes les écritures puis écriture des résultats.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le modèle DTP

- Le Transaction Manager :
 - pilote les sources de données ou de traitements distribués, les Ressources Manager,
 - garantit l'unité d'œuvre (ACIDité) de la transaction.
 - mis en œuvre dans les moniteurs transactionnels, comme Tuxedo, Encina et plus récemment dans certains serveurs d'applications (Microsoft Transaction Server, Oracle Application Server, etc.).
- Le Ressource Manager :
 - utilisé dans les principaux SGBD relationels (Oracle, Sybase, DB2) et dans certains Middlewares orientés messages (MOM).
- Limitations du modèle :
 - la transaction doit être courte car le protocole de validation à deux phases peut présenter des anomalies en cas de pannes durant la phase 1 en bloquant le RM.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le poste client : applications windows

- exécution du client développé grâce un L4G sur le poste local,
 - utilisation des composants de traitement via le réseau avec le middleware (DCOM ou Corba),
 - intégration avec les applications bureautiques via OLE.
- ⇒ Inconvénients : nécessité de disposer du protocole DCOM sur les clients, DCOM peut se trouver bloquer par les firewall, coût élevé de déploiement des applications Windows.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le poste client : applications ou applet Java

- A :
 - similaires aux applications de windows,
 - fonctionne sur le middleware fourni par le serveur d'applications ou plus simplement sur RMI,
 - auto-déploiement de l'application à travers un navigateur.
- ⇒ Inconvénients : nécessite de disposer de la bonne version de la machine virtuelle java et de ses bibliothèques de composants, pas d'intégration vers des outils bureautiques, lenteur des applications.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le poste client : applications WEB

- utilisation du protocole HTTP entre le client et le serveur,
 - traitement effectué au niveau du serveur WEB et donc pas de problème de déploiement,
 - application centralisée et donc plus facile à maintenir,
 - possibilité d'intégrer des applications natives et des documents statiques,
 - développement d'applications « universelles », indépendantes du poste client.
- ⇒ Inconvénients : richesse et ergonomie de l'IHM laissent à désirer, limites inhérentes au langage HTML mais ceci sera compensé par le langage DHTML.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le composant

- Un objet qui adhère à un modèle, il supporte un ensemble de méthodes normées qui lui permettent :
 - d'être exécuté dans un serveur d'applications conforme au modèle en supportant le mode transactionnel, la sécurité, la concurrence d'accès, l'indépendance à la localisation, etc.
 - d'être facilement intégré à un AGL en phase de développement grâce aux méthodes auto-descriptives d'introspection.
- Les méthodes normées se regroupent logiquement sous des interfaces permettant aux composants d'être indépendants de leur implantation.
- Il existe essentiellement deux modèles :
 - Le modèle COM (Component Object Model), issu de Microsoft et du monde OLE, désormais appelé ActiveX. Il est indépendant du langage.
 - Le modèle EJB (enterprise JavaBean), spécification multi-éditeurs dont Sun et IBM sont à l'origine. Il repose sur le langage Java.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Les serveurs d'applications

- Il s'agit d'un cadre d'exécution pour des composants obéissant à un modèle (COM ou EJB), c'est-à-dire un ensemble de services permettant la bonne exécution des composants :
 - les services de base ;
 - l'administration, l'exploitation ;
 - la sécurité ;
 - les passerelles vers l'existant ;
 - la persistance ;
 - les impressions.
- Quatre services de base :
 - l'accès aux composants,
 - l'optimisation de l'accès aux ressources locales et distantes,
 - la gestion transactionnelle,
 - la répartition de charge.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le standard J2EE (1/3)

- **Java 2 Enterprise Edition :**
 - Un ensemble de standards.
 - Définition des modes d'accès à un annuaire, à une base de données.
 - Définition du mode de dialogue entre les machines virtuelles Java.
 - Définition des interfaces qu'un composant doit présenter pour être réutilisable.
- **Objectifs :**
 - Rendre interopérables les composants développés dans le cadre des architectures distribuées.
 - Conception d'architectures multi-niveaux associant :
 - une base de données,
 - un serveur d'applications,
 - et des serveurs Web.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le standard J2EE (2/3)

- Objectifs (suite) :
 - Un composant conforme aux services techniques J2EE doit pouvoir s'exécuter dans n'importe quel environnement estampillé "J2EE".
- La réalité :
 - De nombreux logiciels exploitent une partie des services de J2EE.
 - Peu parmi eux ont passé la totalité des tests de conformité J2EE.
 - Le rattachement d'un éditeur à la norme J2EE nécessite souvent de nombreux mois de travail.

⇒ Implantation partielle de la norme J2EE.
- Tous les logiciels s'appuyant sur un serveur d'applications est concerné par la norme J2EE.

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Le standard J2EE (3/3)

- J2EE, Java et les EJB :
 - J2EE capitalise les technologies poussés par SUN : JAVA, JSP.
 - Les EJB représentent un modèle de composants. Ils représentent donc l'un des points clefs de la technologie J2EE.
- Qui pilote J2EE :
 - Les briques de base ont été élaborées par Sun.
 - Il est possible aux éditeurs de soumettre des mises à jour des spécifications suivant un mécanisme bien défini.
 - La force de J2EE tient à son adoption par les grands éditeurs tels que BEA et IBM.
- Version actuelle de J2EE : Java EE 5
- Documentation :

<http://java.sun.com/javaee/5/docs/tutorial/doc/>

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

Quelques serveurs d'applications

- BEA WebLogic Server (payant).
 - Version 9.2
 - Site Web : <http://www.bea.com/>
- IBM WebSphere (payant).
 - Version 6.1
 - Site Web : <http://www-306.ibm.com/software/websphere/>
- JBoss (gratuit).
 - Version 4.0.5
 - Site Web : <http://www.jboss.org/>
- Jonas (gratuit).
 - Version 4.8.4
 - Site Web : <http://jonas.objectweb.org/>
- Sun One Application Server (gratuit).
- Inprise Application Server (payant).
- Oracle Application Server (gratuit pour une utilisation privée).

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

JOnAS - Java Open Application Server

- Généralités :
 - version actuelle (01/10/2004) : 4.8.4,
 - implémente l'ensemble des fonctionnalités J2EE 1.4.
- Fonctionnalités :
 - intégration native des conteneurs de servlet Tomcat et Jetty,
 - support des services WEB : AXIS, SOAP,
 - intégration du support des environnements de développement : JBuilder, Eclipse.
- Caractéristiques :
 - très grande robustesse, il supporte la charge.
 - utilisé dans de très nombreuses applications : e-commerce, portail, administration des serveurs, intranet, systèmes de réservation, etc.
 - sur différents systèmes d'exploitation Unix, Linux, Windows, AIX, HP-UX, etc.
 - fonctionnant avec de nombreuses bases de données ...

Plan

Introduction

 Architectures
transactionnelles

 Approche à base de
composants

 Serveurs
d'applications

JBoss

- Généralités :
 - créé par le français Marc Fleury (un ancien de Sun),
 - distribué sous licence GPL depuis mars 1999,
 - spécialisé dans la technologie EJB,
 - soutenu par une communauté d'environ 500 développeurs dans le monde, dont un noyau dur de 50 personnes.
- Une architecture modulaire :
 - possibilité de charger dynamiquement un EJB sans redémarrer le serveur,
 - totalement conforme aux spécifications J2EE,
 - exploite les EJB, via un "conteneur" dénommé JBoss Server,
 - les autres fonctions J2EE comme les JSP/servlets sont intégrées via le moteur de servlets open source Tomcat pour Apache.