

Applications Réparties

Département Sciences Informatiques

SI 4^{ème} année

Applications Réparties ?

- ✓ Ensemble de processus (objets, agents, acteurs) qui:
 - Communiquent entre eux via un réseau
 - Evoluent de manière cohérente
 - Remplissent une fonction identifiable
 - Ne sont pas forcément interdépendants

Motivations et Modèles

- ✓ Une évolution logique
 - Généralisation des équipements communicants
 - Interconnexion et haut débit généralisés
 - Répond aux problématiques
 - de passage à l'échelle
 - de tolérance aux pannes
 - d'évolutivité
- ✓ Modèles de programmation:
 - Synchrones (RPC, RMI...)
 - Asynchrone (Evènements)
 - *Ressources partagées (mémoire partagée, systèmes de fichiers répartis)*
 - *Code mobile*
 - *Peer to peer*
 - ...

Programme du Cours

- ✓ Du Web aux Web Services (S. Lavirotte)
 - Couche Transport (exemple HTTP et Serveur Web)
 - Introduction aux Web Services
 - Généralités
 - Architecture
 - Appels de procédures distants: RPC (exemple SOAP)
 - Interface publique de service (exemple WSDL)
- ✓ Bus Logiciel (M. Blay-Fornarino, A.-M. Pinna-Dery)
 - Introduction au Bus Logiciel & RMI
 - RMI et Sécurité (JAAS)
 - Introduction à Corba
 - RMI Corba IIOP
- ✓ Nommage (M. Blay-Fornarino, S. Lavirotte)
 - Annuaire et JNDI
 - UDDI
- ✓ Événements (M. Blay-Fornarino, S. Lavirotte, A.-M. Pinna-Dery)
 - JMS
 - Introduction à UPnP

De Nombreux Intervenants

- ✓ Pour les cours:
 - Mireille Blay-Fornarino
 - Stéphane Lavirotte
 - Anne-Marie Pinna-Dery

- ✓ Pour les Travaux Dirigés:
 - Mireille Blay-Fornarino
 - Nicolas Ferry
 - Stéphane Lavirotte
 - Dino Lopez-Pacheco
 - Anne-Marie Pinna-Dery

Du Web aux Web Services

Stéphane Lavirotte
Dino Lopez-Pacheco
Nicolas Ferry

(*) D'après les cours de Jean-Yves Tigli, Gaëtan Rey,
Stéphane Lavirotte, Michel Riveill, Sacha Krakowiak,
Didier Donsez, et Keith Ross

et

<http://abcdrfc.free.fr/>



Du Web aux Web Services

✓ Présentation du Cours

- Cours 1 et 2 :
 - Couche Transport* (exemple HTTP et Serveur Web)
- Cours 3 à 5:
 - Introduction aux Web Services
 - Généralités
 - Architecture
 - Appels de procédures distants: RPC (exemple SOAP)
 - Interface publique de service (exemple WSDL)
- Cours 11:
 - Nommage (UDDI)
 - Événements (Introduction UPnP)
 - Conclusion

Du Web aux Web Services

✓ Travaux Dirigés:

- TD n°1 :
 - Conception d'un serveur HTTPD (liaison avec le module Internet et Réseau)
- TD n°2 :
 - Administration HTTPD / Apache
- TD n°3 :
 - Web Services avancés sous .Net (C#)
- TD n°4 :
 - Interopérabilité : WS avec gSOAP (C++)
- TD n°5 :
 - Interopérabilité : WS avec Apache Axis (Java)
- TD n°11 :
 - UDDI sous .Net (C#)

Quelques Définitions

Le Jargon du Web

- ✓ Page Web:
 - Pointés par une URL
 - La plupart des pages WEB se composent de:
 - Une page HTML de base,
 - Différentes références à des « objets »

- ✓ L'agent utilisateur pour le Web s'appelle un "browser" (navigateur en français)
 - Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari, Google Chrome, ...

- ✓ Un serveur pour le Web s'appelle un serveur Web :
 - Apache, Microsoft Internet Information Server (IIS), ...



URx: Uniform Resource ...

- ✓ Une URL (Uniform Resource Locator) a au moins trois champs (protocole, adresse, emplacement) :
 - Le protocole: *http* suivi de :
 - Le nom complet de la ressource: données d'authentification à la ressource (*// login : password @ nom domaine : port*)
 - Nom de la ressource sur le service auquel on est connecté
 - Données supplémentaires optionnelles transmises
- ✓ Un URN (Uniform Resource Name)
 - Identifie une ressource par un nom dans un espace de nommage (identifie la ressource et pas sa localisation)
 - urn:NID:NSS
- ✓ Plus généralement un URI (Uniform Resource Identifier)
 - Peut être une URL ou un URN

Le Protocole HTTP

HyperText Transfert Protocol

HTTP 1.0 : RFC 1945

HTTP 1.1 : RFC 2616

Introduction

Les Principes et Éléments de base
du Protocole

HyperText Transfer Protocol

✓ *HTTP : HyperText Transfert Protocole*

- Un des protocoles les plus courants sur Internet
 - Un protocole omni-présent: de IT à Embedded
- Il est utilisé pour la navigation sur les sites Web
 - protocole de rapatriement des documents
 - protocole de soumission de formulaires

✓ Il en existe trois versions :

- 0.9 (1991) : complètement obsolète
- 1.0 (février 1997), de nos jours très rarement utilisée
- 1.1 (octobre 2000). Les principaux changements entre les v1.0 et v1.1 sont l'ajout de 2 types de requêtes ainsi que la possibilité d'héberger plusieurs sites Web sur un même serveur dans la version 1.1.

Un Modèle Client - Serveur

- ✓ **Modèle Client / Serveur**
 - client: « browser » qui demande, reçoit et affiche des documents Web.
 - server: serveur Web qui envoie des documents en réponse aux requêtes des clients.

- ✓ **L'échange entre le client et le serveur se fait en mode texte.**
 - Le charset généralement utilisé est l'US-ASCII sur 8 bits.
 - Il est cependant possible que cet encodage soit modifié selon le client ou le serveur.



Principe de Fonctionnement de HTTP

- ✓ HTTP : TCP transport service
 - Le client initialise une connexion TCP/IP (voir sockets) sur le serveur et le port 80.
 - Le serveur accepte la connexion du client et fournit un port de communication (utilisateur).
 - Les messages http (messages au protocole de l'application) sont échangés entre le client http et le serveur http.
 - Enfin, la connexion TCP/IP est fermée.

- ✓ HTTP est "stateless"
 - Le serveur ne maintient pas d'information sur les requêtes passées du client.

Zoom sur un Exemple de Communication

1. Le client http initialise une connexion TCP sur le serveur http *www.unice.fr*. (sur le port 80)
2. Le serveur http *www.unice.fr* en l'attente de connexions sur le port 80, accepte la demande de connexion du client
3. Le client http envoie un message de requête (contenant l'URL) au travers le socket de communication TCP.
4. Le serveur http reçoit le message de requête, compose le message de réponse contenant les objets demandés et renvoie le message au travers le socket de communication.
5. Le client http reçoit le message de réponse contenant le fichier HTML et l'affiche.
6. Le serveur http ferme la connexion.
7. En « parsant » le fichier HTML, le client http trouve 10 références à des objets jpeg. Les étapes 1 à 6 sont répétées pour chaque référence aux 10 objets jpeg.

Persistances des Connexions

- ✓ Connexions non-persistentes
 - HTTP/1.0
 - Le serveur « parse » la requête, répond et ferme la connexion.
 - Ralentit la récupération de la page complète
 - Mais la plupart des « browsers » 1.0 utilise des connexions parallèles
- ✓ Connexions persistentes
 - Par défaut pour HTTP/1.1
 - Durant une même connexion TCP, le serveur « parse » une requête, répond puis recommence ..
 - Le client envoie des requêtes pour tous les objets référencés aussitôt qu'il reçoit la page HTML de base.
 - Accélère la récupération de la page complète

En résumé

- ✓ Le fonctionnement de HTTP est très simple pour HTTP/1.0
 - connexion
 - demande (GET) d'un document
 - renvoi du document (status=200) ou d'une erreur
 - Déconnexion

- ✓ Cependant
 - Dialogue plus complexe en cas d'authentification
 - Optimisation : une série de plusieurs requêtes sur une connexion [Connexion « KeepAlive » de HTTP/1.1 (RFC 2068)]

Dialogue HTTP

✓ Dialogue

- en mode caractères ASCII (7 bits)
 - telnet www.sun.com 80

✓ Types de dialogue

- Récupération d'un document
 - méthode GET
- Soumission d'un formulaire
 - méthodes GET ou POST
- Envoi de Document et Gestion de Site
 - méthodes PUT, DELETE, LINK, UNLINK
- Gestion de proxy/cache
 - méthode HEAD (récupération des informations sur le document)

Structure

- ✓ Les requêtes et les réponses sont bâties sur le même modèle

```
{Ligne d'introduction}{SEP}  
{En-têtes séparées par des {SEP}}  
{SEP}{SEP}  
{Corps}
```

- ✓ Le seul élément capable de différencier une requête d'une réponse, c'est la *Ligne d'introduction*.



Format de la Requête



Méthodes de la Requête

- ✓ **GET**
 - demande pour obtenir des informations et une zone de données concernant l'URI
- ✓ **HEAD**
 - demande pour seulement obtenir des informations concernant l'URI
- ✓ **POST**
 - envoie de données (contenu du formulaire vers le serveur, requête SOAP ...). Ces données sont situées après l'entête et un saut de ligne
- ✓ **PUT**
 - enregistrement du corps de la requête à l'URI indiqué
- ✓ **DELETE**
 - suppression des données désignées par l'URI



Méthodes de la Requête

✓ OPTIONS

- demande des options de communication disponibles

✓ TRACE

- retourne le corps de la requête intacte (débugage)

✓ LINK / UNLINK

- association (et désassociassion) des informations de l'entête au document sur le serveur

✓ **Nouvelles extensions de WebDAV**

- PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK

✓ **Nouvelles extensions HTTP/U HTTP/MU**

- NOTIFY, ... (UPnP)

Champs d'Entête

- ✓ Ils permettent la transmission d'informations complémentaires sur la requête, et le client lui-même.
- ✓ Ces champs agissent comme "modificateurs" de la requête, utilisant une sémantique identique à celle des paramètres passés par un appel d'une méthode de langage de programmation de haut niveau.



Format de la Réponse



Statuts des Réponses HTTP (RFC2068)

- ✓ **1xx Information**
 - 100 : Continue (le client peut envoyer la suite de la requête), ...
- ✓ **2xx Succès de la requête client**
 - 200: OK, 201: Created, 204 : No Content, ...
- ✓ **3xx Redirection de la Requête client**
 - 301: Redirection, 302: Found, 304: Not Modified, 305 : Use Proxy,
- ✓ **4xx Requête client incomplète**
 - 400: Bad Request, 401: Unauthorized, 403: Forbidden, 404: Not Found
- ✓ **5xx Erreur Serveur**
 - 500: Server Error, 501: Not Implemented,
 - 502: Bad Gateway, 503: Out Of Resources (Service Unavailable)