Technologies du Web

Programmation Web côté serveur

Pierre Senellart (pierre.senellart@telecom-paristech.fr)

Mastère spécialisé *Management et nouvelles technologies*,
16 novembre 2009

Plan du cours

Architecture client-serveur

Rappel

- Un client C (p.ex. un navigateur Web), sur une machine X.
- Un serveur Web S sur une machine Y.
- C se connecte à Y.
- C demande à S une URL, accompagnée de paramètres.
- S répond à C en lui renvoyant p.ex. une page Web. Cette page peut être un document statique (p.ex. un fichier HTML) ou une page dynamique produite par un programme (p.ex. un script PHP).
- S ferme la connexion à C.

Serveurs Web

Serveur	Part	Distribution
Apache	60%	libre, Windows, Mac OS, Linux, Unix
Microsoft IIS	30%	avec certaines versions de Windows
lighthttpd	2%	libre, Windows, Mac OS, Linux, Unix
Non identifiable	8%	

- Parts de marché : d'après des études par Opera et Netcraft, chiffres précis ne signifient pas grand chose.
- De nombreuses grandes entreprises de logiciel ont soit leur propre logiciel, soit leur propre version modifiée d'Apache (notamment, GFE/GWS pour Google).
- lighthttpd est (comme son nom l'indique !) plus léger (c'est-à-dire, moins riche en fonctionnalités mais plus rapide dans certains contextes) qu'Apache.
- Les versions de Microsoft IIS venant avec les Windows grands publics sont très limitées.

HyperText Transfer Protocl

- Protocole de communication à la base du World Wide Web.
- Requête du client :

```
GET /MarkUp/ HTTP/1.1
Host: www.w3.org
```

- Réponse du serveur :

```
HTTP/1.1 200 OK
...
Content-Type: text/html; charset=utf-8

<!DOCTYPE html ...>
  ...
```

- Deux principales méthodes HTTP : GET et POST.
- Des en-têtes additionnels, dans la requête et dans la réponse, pour diverses fonctionnalités.
- Possibilité de passer des paramètres dans la requête (clef/valeur).

GET

- Type de requête la plus simple.
- Les paramètres (éventuels) sont passés à la fin de l'URL, après un ' ?'
- Ne convient pas quand il y a beaucoup de paramètres ou que leurs valeurs sont très longues.
- Méthode utilisée quand on visite directement une URL, quand on suit un lien, et pour certains formulaires.

Exemple (Recherche Google)

URL : `http://www.google.fr/search?hl=fr&q=hello`

Requête HTTP GET correspondante :

```
GET /search?hl=fr&q=hello
Host: www.google.fr
```

POST

- Méthode utilisée uniquement en soumettant un formulaire.

Exemple

```
POST /php/test.php HTTP/1.1
Host: www.w3.org
Content-Type: application/x-www-form-urlencoded
Content-Length: 100

type=recherche&titre=Le+Dictateur&format=long&pays=US
```

Encodage des paramètres

- Par défaut, les paramètres sont passés (en GET ou en POST) sous la forme : `name1=valeur1&name2=valeur2`, et les caractères spéciaux (caractères accentués, espaces. . .) sont remplacés par des codes comme +, %20. Cette manière de passer les paramètres est nommée `application/x-www-form-urlencoded`.
- Pour la méthode POST, on peut aussi choisir un encodage plus lourd (plusieurs lignes par paramètre), similaire à la façon dont les e-mails sont construits ; c'est surtout utile pour passer de grandes quantités d'information. On parle d'encodage `multipart/form-data`. Seule méthode utilisable pour transmettre des fichiers avec un `<input type="file">`.

Codes de statut

- La réponse HTTP commence toujours par un code de statut sur trois chiffre, suivi d'un message lisible par un être humain (p. ex., 200 OK).
- Le premier chiffre indique la classe de la réponse :
  1. Information
  2. Succès
  3. Redirection
  4. Erreur client
  5. Erreur serveur

Codes de statut les plus courants

200	OK
301	Redirection permanente
302	Redirection temporaire
304	Pas de modification
400	Requête invalide
401	Non autorisé
403	Interdit
404	Page non trouvée
500	Erreur serveur

Hôtes virtuels

- Différents noms de domaines peuvent pointer sur la même adresse IP, c'est-à-dire la même machine physique (p. ex., `www.google.fr` et `www.google.com`)
- Quand on contacte une machine en TCP/IP, c'est par son adresse IP
- Donc pas de moyen a priori de connaître le nom de domaine précis que l'on veut joindre
- Pour pouvoir servir du contenu différent suivant le nom de domaine (= hôte virtuel : en-tête `Host:` de la requête (seul en-tête vraiment requis de la requête)

Exemple

```
GET /search?hl=fr&q=hello
Host: www.google.fr
```

Type de contenu

- Le navigateur doit se comporter différemment suivant le type de contenu renvoyé : afficher une page Web avec le moteur de rendu, afficher une image, charger une application externe, etc.
- Classification MIME des types de contenu (p. ex., `image/jpeg`, `text/plain`, `text/html`, `application/xhtml+xml`, `application/pdf` etc.
- Pour une page HTML, ou du texte, le navigateur doit savoir également quel est le jeu de caractères utilisé (attention : a précédence sur l'information contenue dans le document lui-même)
- Également renvoyé : la longueur de contenu (permet d'afficher une barre de progression et de savoir quand le serveur a terminé)

Exemple

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 3046
```

Identification du client et du serveur

- Client web et serveur Web peuvent s'identifier via une chaîne de caractères plus ou moins codifiée
- Utile parfois pour servir du contenu différent à différents navigateurs, détecter des robots...
- ... mais n'importe quel client peut se faire passer pour n'importe quel autre !
- Confusions historiques sur les noms : tous les navigateurs courants s'identifient comme Mozilla !

Exemple

```
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; fr;
rv:1.9.0.3) Gecko/2008092510 Ubuntu/8.04 (hardy)
Firefox/3.0.3

Server: Apache/2.0.59 (Unix) mod_ssl/2.0.59 OpenSSL/0.9.8e
PHP/5.2.3
```

Authentification

- HTTP prévoit un mécanisme de protection d'un site Web par identifiant et mot de passe
- Attention : (la plupart du temps) le mot de passe circule non chiffré
- Privilégier HTTPS pour transmettre des mots de passe sensibles
- Méthode la plus courante : login et mot de passe encodé en Base64 (encodage réversible !)

Exemple

```
GET ... HTTP/1.1
Authorization: Basic dG90bzp0aXRp
```

Négociation de contenu

- Un client Web peut préciser au serveur Web :
  - le type de contenu qu'il prend en charge (textes, images, contenus multimédias), avec des indicateurs de préférences
  - les langues préférées par l'utilisateur
- Le serveur Web peut ainsi proposer des types de fichier différents, dans des langues différentes.
- En pratique, la négociation de contenu selon les langues fonctionne, et est utilisée, mais la négociation de contenu selon les types ne fonctionne pas car certains navigateurs déclarent des types préférés folkloriques.

Exemple

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
```

Cookies

- Informations, sous la forme de clés/valeurs, qu'un serveur Web demande à un client Web de conserver et de retransmettre à chaque requête HTTP (pour un nom de domaine donné).
- Sert à conserver des informations sur un utilisateur au fur et à mesure de la visite d'un site Web, entre deux visites, etc. : panier électronique, identifiant, etc.
- En pratique, stocke le plus souvent uniquement un identifiant de session, qui est relié, côté serveur, à toutes les informations sur la session (connecté ou non, nom d'utilisateur, données...)

Exemple

```
Set-Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2;
path=/; domain=.amazon.de;
expires=Fri Oct 17 09:35:04 2008 GMT

Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2
```

Téléchargement conditionnel

- Le navigateur peut demander le téléchargement d'une page que si elle a été modifiée depuis une certaine date.
- Peu souvent applicable, le serveur donnant peu souvent une date de dernière modification (difficile à obtenir pour du contenu géré dynamiquement !).

Exemple

```
If-Modified-Since: Wed, 15 Oct 2008 19:40:06 GMT

304 Not Modified
Last-Modified: Wed, 15 Oct 2008 19:20:00 GMT
```

URL d'origine

- Quant un navigateur Web suit un lien, ou soumet un formulaire, celui-ci transmet l'URL d'origine.
- Même si ce n'est pas sur le même serveur !

Exemple

```
Referer: http://www.google.fr/
```

Configuration d'Apache

- Fichier de configuration global pour le serveur Apache
- Ou fichiers `.htaccess` dans chaque répertoire

Exemple

```
AddDefaultCharset utf-8 # jeu de caractères

# Redirection
RedirectMatch permanent ^/toto(.*) http://www.example.com/titi

# Authentification
AuthName "Entrez votre login/mdp"
AuthType Basic
AuthUserFile /home/pierre/divers/htpasswd
```

Plan du cours

Rôle des programmes côté serveur

- Le Web, ce n'est pas qu'un ensemble de documents HTML statiques !
- Les programmes côté serveur permettent :
  - de traiter des soumissions de formulaire ;
  - d'afficher de manière uniforme l'ensemble des pages d'un site ;
  - de proposer des applications interactives ;
  - de permettre à l'utilisateur d'ajouter ou modifier du contenu ;
  - etc.

Langages de programmation

- CGI (Common Gateway Interface) : interface normalisée permettant de faire communiquer le serveur Web avec un programme s'exécutant sur le serveur ;
- CGI permet d'utiliser n'importe quel langage de programmation (compilés comme C, C++, Java, ou interprétés comme Perl, Python, Ruby, etc.) pour écrire des langages côté serveur.
- Mais certains langage sont plus adaptés au développement Web :
  - comportent des fonctions spécialement dédiées à HTTP, HTML, etc. ;
  - s'intègrent de manière plus efficace et plus pratique avec le serveur Web (moyennant des extensions logicielles) ;
  - ont une syntaxe spécialement conçue, qui mêle code HTML envoyé tel quel et instructions de programmation interprétées.
- Quelle que soit la technologie utilisée, le code du programme n'est pas accessible par le navigateur Web, seulement le résultat de son exécution.

Langages côté serveur

PHP : un des langages les plus populaires, s'intègre très facilement avec Apache (libre)

ASP et ASP.NET : destiné à être utilisé avec IIS (Microsoft, commercial)

ColdFusion (Adobe, commercial)

JSP (Java Server Pages) : permet de mêler instructions Java et code HTML ; nécessite un serveur d'applications Java (p. ex., Tomcat) en plus d'Apache (Sun, gratuit voire libre)

Servlets Java : véritables programmes Java, plutôt pour les applications complexes côté serveur avec peu d'interaction côté client ; nécessite un serveur d'applications Java en plus d'Apache (Sun, gratuit voire libre)

Frameworks d'applications Web

- Les langages présentés ci-avant restent assez basiques et généralistes.
- N'encouragent pas forcément une organisation propre d'un site Web.
- Framework : ensemble d'un langage de programmation, d'une bibliothèque de fonctions, d'outils externes, de bonnes pratiques à suivre...
- Permet d'abstraire la création d'une page Web.
- Suit en général le modèle MVC (voir transparent suivant).
- Inclut parfois la génération de code JavaScript côté client pour créer directement une application Web fortement dynamique (p. ex., validation de formulaire) ; intégration Ajax également.
- Fortement recommandé pour créer des applications complexes... mais également complexe à maîtriser !

Modèle MVC

- Principe de génie logiciel, utilisé dans d'autres domaines
- Particulièrement adapté au cas des applications Web !
- Séparation propre entre :

  Modèle : données manipulées par l'application et fonctions de manipulation de ces données ; réutilisable pour d'autres applications Web

  Vue : présentation des données ; facilement échangeable pour changer l'apparence et la structure du site

  Contrôleur : contrôle la manière dont l'utilisateur interagit, au travers de la vue, avec les données du modèle

Frameworks côté serveur les plus populaires

- ASP.NET : DotNetNuke
- ColdFusion : Model-Glue, Fusebox
- Java : Struts, Spring, JavaServer Faces, Google Web Toolkit
- Perl : Catalyst
- PHP : CakePHP, Symphony, Zend
- Python : Django
- Ruby : Ruby on Rails (a eu beaucoup d'influence !)
- Smalltalk : Seaside

. . . et beaucoup d'autres !

Systèmes de gestion de contenu

- CMS (Content Management System)
- En quelque sorte une version « extrême »des frameworks : plus besoin de programmer, application (presque) prête à l'emploi
- Permet de gérer des documents, des fichiers multimédias, des blogs, des systèmes de commentaires, etc.
- Permettent souvent de développer ses propres extensions, à la manière d'un framework, pour une personnalisation totale
- Les plus courants : Mambo, Joomla!, Drupal, Wordpress, Plone...
- CMS d'un type un peu particulier : Wiki (MediaWiki, TWiki...)
- + applications Web spécialisées : gestion de forums, d'agenda, de serveur mail, etc.

Repérer une technologie côté serveur

Impossible d'avoir accès au code source, donc comment savoir avec quelle technologie côté serveur un site a été réalisé ?

- URL : (`.php` pour PHP, `.jsp` pour JSP, `.asp` pour ASP, `/servlet/` pour une servlet...). Tout cela est configurable, mais bien souvent la configuration par défaut est utilisée.
- Ça peut être écrit en toutes lettres sur le site (souvent pour un CMS)
- Regarder les commentaires dans le code HTML, CSS...
- Organisation des fichiers et répertoires, librairies JavaScript ou CSS chargées, etc.
- Cookies, en particulier identifiants de session

Plan du cours

PHP et HTML

- Script PHP : document HTML (par exemple), dans lequel est incorporé du code PHP.
- Le code PHP est à l'intérieur d'une pseudo-balise `<?php ... ?>` (ou `<? ... ?>`, ou `<?= ... ?>` qui est un raccourci pour `<? echo ... ?>`).

Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
  ...
  <body>
  <h1><?php echo 2+2; ?></h1>
</body>
</html>
```

Introduction au langage

- Un script PHP est une suite d'instructions terminées par des points-virgules.

Exemple (Écrire une phrase avec l'instruction echo)

```
echo 'Ceci apparaîtra dans la page générée.';
```

- Ces instructions contiennent des éléments variables ou constants, peuvent être conditionnées ou encore itérées plusieurs fois.

Notion de variable

- Une variable est le nom d'un récipient destiné à contenir une valeur. Dans un ordinateur, ce récipient est en fait une zone mémoire.
- L'affectation `$var=valeur` est une instruction qui permet de changer la valeur d'une variable `$var` :
  - La valeur de la variable à gauche de `=` est remplacée par la valeur à droite de `=`.
- L'affectation est une instruction dite destructive : l'ancienne valeur de la variable est détruite, écrasée, effacée par la nouvelle valeur !
- Une valeur peut être :
  - un nombre entier ou réel ;
  - une chaîne de caractères définie entre deux guillemets simple (`'`) ou entre deux guillemets double (`"`).

Affectation

- La valeur affectée à une variable peut également être le résultat d'une expression.
- Attention, à droite d'une affectation `=`, les variables utilisées dans l'expression désignent les valeurs qu'elles contiennent.

Exemple

`$Compteur= $Compteur + 1;` a pour effet de mettre le résultat de la somme de la valeur de `$Compteur` avec la valeur 1 dans le récipient `$Compteur`.

Instruction conditionnelle

- Une instruction conditionnelle est de la forme :
  Si *la condition C est vraie* alors *faire liste d'instructions T*
  sinon *faire liste d'instructions E*
- où une condition est une expression et T (resp. E) est un bloc d'instructions.
- En PHP, on utilise la syntaxe suivante :

```
if (C) { T } else { E }
```

Exemple

```
if ($sexe=='f') {
  echo 'Madame';
} else {
  echo 'Monsieur';
}
```

- La condition est généralement un test de comparaison, ou plusieurs tests reliés par des opérateurs logiques.
- Le `else` et son bloc associé sont optionnels.
- Si un bloc ne contient qu'une seule instruction, on pourra omettre les accolades.

Exemple

```
if ($sortie=='q') { echo 'Merci et au revoir!'; }

if ($a > $b) { echo "$a est plus grand que $b"; }
else {
  if ($a==$b) { echo "$a est égal a $b"; }
  else { echo "$a est plus petit que $b"; }
}
```

Boucles

- Boucle : permet de répéter une opération

```
for (I;C;P) { B }
```

- où *I* est une affectation permettant d'initialiser la variable itérée,
- *C* est la condition d'arrêt de la boucle (valeur limite que doit atteindre la variable itérée),
- *P* est le pas de la boucle : c'est une affectation qui permet de modifier à chaque itération la valeur de la variable itérée,
- *B* est le bloc d'instructions itéré.

Exemple

```
for( $i=0 ; $i<10 ; $i=$i+1 ) { echo 'Au secours!'; }

$fact=1;
for( $i=13 ; $i>0 ; $i=$i-1 ) { $fact = $fact*$i; }
```

$_GET et $_POST

- Les paramètres HTTP peuvent être récupérées en PHP grâce au tableaux associatifs `$_GET` et `$_POST`.
- Les valeurs de ce tableau peuvent être des variables simples ou des tableaux indicés : ces derniers sont les paramètres à choix multiples dont on a suffixé le nom de `[]` dans le code HTML.

Exemple

```
echo "<p>Votre login est : " . $_POST["login"] . "</p>";
echo "<p>Vous avez coché les genres : ";
for($i=1;$i<=count($_POST['genre']);$i=$i+1) {
  echo $_POST['genre'][$i] . " ";
}
echo "</p>";
```

Validation

- Un script PHP avec du code PHP incorporé au sein de code HTML n'est pas un document XHTML valide !
- Ce qu'il faut valider, c'est une (des) page(s) HTML produites par le script.
- Possibilité d'indiquer une URL au validateur du W3C. Malheureusement pas utilisable quand l'URL est privée. Possibilité dans ce cas de sauvegarder le fichier HTML produit, et de l'envoyer au validateur du W3C comme fichier local.
- Il n'y a pas de « validateur » PHP, mais les erreurs de syntaxe causeront des erreurs à l'exécution du script.
- Rien ne change pour les CSS, on peut les valider à part.

Plan du cours

SGBD

- SGBD : Système de Gestion de Bases de Données
- Fournit des méthodes efficaces pour gérer des données qui répondent à une structure (un schéma) précis.
- Rechercher des données : requêtes
- Ajouter, supprimer, modifier des données : mises à jour
- Traite de manière rapide de grandes quantités de données.
- Gérer des transactions (ne pas donner à deux clients la même place dans un train !)

Modèle relationnel

- Modèle le plus répandu et le plus classique.
- Les données sont organisées en des tables, chacune des colonnes représentant un attribut des données.

Exemple

Prénom	Nom	Date de naissance
Jean	Dupont	1967-08-07
Pascale	Dupuis	1981-09-12
Alfred	Lambert	NULL

- Chaque attribut (colonne) est typé.
- SQL (Structured Query Language) : langage standard de requête et de mise à jour des données (petites variantes suivant les SGBD).

SGBD relationnels les plus connus

Oracle	commercial, le plus utilisé en entreprise
IBM DB2	commercial
Microsoft SQL Server	commercial
Microsoft Access	commercial, pauvre en fonctionnalités
MySQL	libre, le plus utilisé sur le Web
PostGreSQL	libre, plus abouti et plus complexe que MySQL

Types de données

(Types MySQL)

- **INT :** entier (`42`)
- **REAL :** nombre en virgule flottante (`3.14159`)
- **VARCHAR(N) :** chaîne de caractères ayant au plus N caractères ; les valeurs sont délimitées par des apostrophes (`'Ceci est une chaîne'`).
- **TEXT :** longue chaîne de caractères, sans limite de taille
- **DATE :** date (`2005-11-08`)
- **TIME :** temps (`09:30:00`)

NULL

- NULL : valeur spéciale
- Dénote l'absence de valeur.
- Différent de `0`, de `''`...
- Une comparaison normale (`=`, `<>`) avec `NULL` renvoie toujours FAUX.
- `IS NULL`, `IS NOT NULL` peuvent être utilisées pour tester une valeur.
- Chacune des colonnes doit être déclarée comme acceptant ou non la valeur `NULL`.

SQL

- Langage standardisé pour interagir avec un SGBD.
- Exemple de requête :

```
SELECT a.name, COUNT(*)
FROM actors a, casting m
WHERE a.id=m.actor_id
GROUP BY a.name
```

- Exemple de mise à jour :

```
UPDATE availabilities
SET nb_available=nb_available-1
WHERE train_id=1234
```

Bases de données pour le Web

- SGBD très utiles pour beaucoup d'applications Web (annuaires, catalogues, réservations, forums de discussion, blogs, etc.)
- Possibilité d'accéder aux bases de données plus ou moins simplifiée suivant les langages :
  - En PHP, bibliothèques de fonctions propres à chaque SGBD (p. ex., `mysql` ou `mysqli` pour MySQL).
  - En JSP, bibliothèque JDBC avec les scriptlets et balises `<sql:*>` avec la JSTL, pour accéder à n'importe quel SGBD.

Exemple : utilisation de MySQL avec PHP

```
if(!mysql_pconnect("server","login","password"))
  { echo "<p>Desolé, connexion impossible</p>"; exit; }
if(!mysql_select_db('database'))
  { echo "<p>Desolé, accès à la base impossible</p>"; exit; }

$resultat= mysql_query("SELECT * FROM Films");
if($resultat) {
   while($film=mysql_fetch_assoc($resultat)){
     echo "<p>".$film["Titre"]." est paru en ".
       $film["Annee"]." </p>";
  }
} else {
   echo "<p>Erreur dans l'éxécution de la requête.</p>";
   echo "<p>Message de MySQL: ".mysql_error()."</p>";
}
```

Plan du cours

Outils

- Outils HTTP : Netcat, Wget, Curl
- Firebug pour analyser du trafic HTTP
- N'importe quel éditeur de textes pour la plupart des langages côté serveur
- Un serveur Web (p. ex., Apache)
- L'interpréteur du langage côté serveur (p. ex., PHP et mod_php pour Apache)
- Pour JSP, Servlets : un serveur d'applications Java (p. ex., Tomcat)
- Un SGBD (p. ex., MySQL) ; sous Windows, EasyPHP regroupe Apache+PHP+MySQL
- La plupart du temps, un moyen de transférer les programmes vers le serveur Web distant (SSH, FTP, application Web...)

Références

HTTP
- http://www.faqs.org/rfcs/rfc2616.html
- http://www.ietf.org/rfc/rfc2109.txt
- http://www.ietf.org/rfc/rfc2965.txt

PHP
- http://www.php.net/
- *Pratique de MySQL et PHP*, Philippe Rigaux, O'Reilly

MySQL : http://dev.mysql.com/doc/

Plan du cours

Application

Écrire et tester des applications Web en PHP :

- un programme qui affiche la somme de deux entiers passés en paramètres GET ;
- un programme qui dit si une année passée en paramètre GET est bissextile ;
- un programme qui construit la table de multiplication (en HTML !) correspondant à un entier passé en paramètre GET.