Techniques de programmation Web côté serveur

David Gross-Amblard - Philippe Rigaux

30 octobre 2002

Conservatoire national des arts et métiers

Transparents disponibles sur http://dept25.cnam.fr:8080/PROJET3I

Contenu

- Programmation côté serveur
- Interface CGI
- Langage PHP
  - Généralités
  - Programmation avancée
  - Application

Logiciels disponibles

Site central

- `http://www.php.net`

Code et doc en ligne

- `http://www.phpindex.com`
- `http://www.phpbuilder.com`
- `http://www.manucorp.com`, etc.

Pour jouer à la maison

- `http://www.easyphp.org` (Windows)
- Paquetages Apache et Php (Linux)

Bibliographie

- P. Rigaux, Pratique de Mysql et Php, O'Reilly (nouvelle édition en préparation)
- En général, O'Reilly sur Apache, Mysql, Php
- ...sinon, de tout et n'importe quoi

Programmation côté serveur

Objectif

Pages web dont le contenu varie

- Engendrées par des programmes
- En fonction des demandes de l'utilisateur
- Côté navigateur : javascript
- Coté serveur : CGI, PHP, ASP, JSP

$\Rightarrow$c'est le serveur qui travaille, pas le client

Passage d’information

Le navigateur peut envoyer de l’information

- dans l’URL utilisée
  $\Rightarrow$`http://www.google.com/?q=panda`
- dans l’en-tête HTTP

Source des données

Les formulaires HTML

- conteneur `<FORM ACTION=u METHOD=m>` et `</FORM>`
- `u`: une URL traitant les informations
- `m` : `GET` ou `POST`

Source des données

Une balise de saisie pour chaque donnée

```
<INPUT TYPE=t NAME=n VALUE=v>
```

- `t` : type (text, radio,submit,...)
- `n` : nom de l'information transmise
- `v` : valeur par défaut

Exemple

```
...<BODY>
<FORM ACTION="calcul.cgi" METHOD=GET>
Combien ? <INPUT TYPE=TEXT NAME=montant VALUE=1>
<BR>
En francs <INPUT TYPE=RADIO NAME=devise VALUE=fr>
<BR>
En euros <INPUT TYPE=RADIO NAME=devise VALUE=eur>
<BR>
Et hop : <INPUT TYPE=SUBMIT VALUE="Convertir">
</BODY>
```

démo

Transmission de l'information

- paires `nom=valeur`
- codage des caractères spéciaux
  ex. `"bonjour monsieur"`
  $\Rightarrow$`"bonjour%20monsieur"`
- Méthode `GET`
  - Information dans l'URL
  - `n1=v1 & n2=v2 & ...`
- Méthode `POST`
  - Dans l'en-tête `HTTP`

Méthode GET

- Avantage
  - Utilisable dans une URL

```
<A HREF=http://www.google.com?q=panda>
    les pandas</a>
```

- Inconvénients
  - URL très longues
  - Informations visibles (sécurité)

Méthode POST

- Avantages
  - Transmission de grosses quantités d'informations
    ex. fichiers (images, sons,...)
  - Invisible

Programmation CGI

Common Gateway Interface

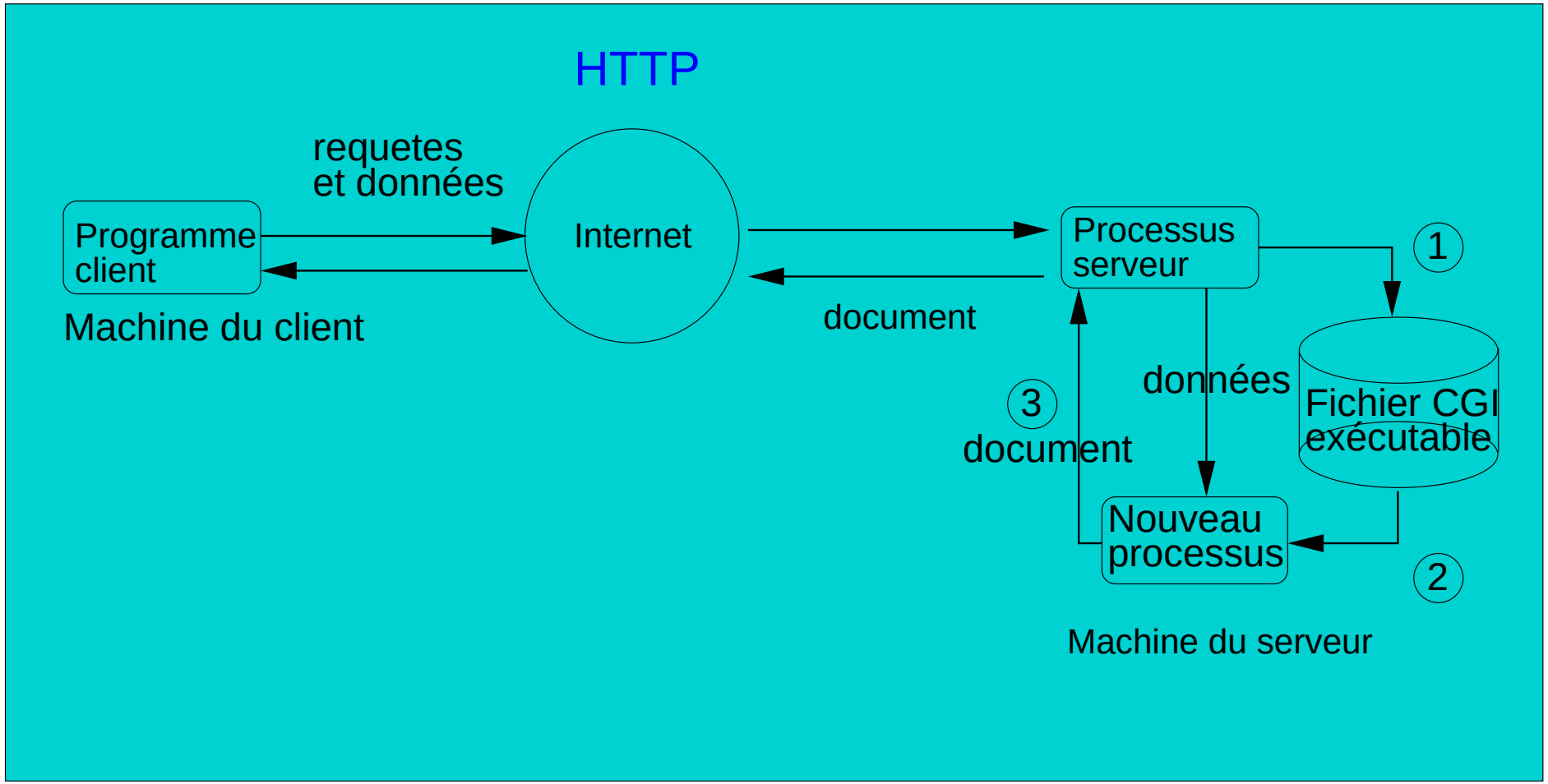
Protocole d’interconnexion entre serveur et n’importe quel programme

- Certaines pages web déclarées comme CGI (exécutables)
- On peut utiliser son langage favori !

Déroulement d’un CGI

1. Le serveur reçoit les données
2. Il lance le processus correspondant et transmet les données par l’entrée standard
3. Le processus produit le résultat sur sa sortie standard
4. La page ainsi formée est envoyée, le processus termine

Déroulement d'un CGI



Passage des données

Les informations circulent comme des variables prédéfinies

- `QUERY_STRING` : les informations transmises
- `HTTP_USER_AGENT` : le navigateur utilisé
- `REMOTE_ADDR` : nom du client (ou adresse IP)
- etc.

Exemple de CGI en C

```
int main(char** argv){
  char* BROWSER=getenv("HTTP_USER_AGENT");
  char* QUERY_STRING=getenv("QUERY_STRING");
  printf("Content-type: text/html\n");
  printf("\n");
  printf("<HTML>\n");
  printf("<BODY>\n");
  printf("<UL>\n");
  printf("<LI>Votre navigateur : \%s\n",BROWSER);
  printf("<LI>Données transmises : \%s\n",QUERY_STRING);
  printf("</UL>\n");
  printf("</BODY>\n");
  printf("</HTML>\n");
}
```

démo

En bref

Avantages

- Solution générale
- Langage favori

Inconvénients

- Bas niveau : il faut analyser `QUERY_STRING`
- Peu efficace : un processus indépendant à chaque appel !

Programmation PHP

PHP (Hypertext Preprocessor)

- Langage de script
- Dans le document HTML : `<?php ... ?>`
- Fonctions (un peu modulaire)
- Objets
- Décodage des données transmises automatique
- Intégré au serveur web (Apache)

Exemple précédent, en PHP

```
<HTML>
<BODY>
<UL>
<LI>Votre navigateur : <?php echo $HTTP_USER_AGENT; ?>
<LI>Données transmises :
<?php echo $QUERY_STRING; ?>
<LI>La question posée :
<?php echo $question; ?>
</UL>
</BODY>
</HTML>
```

démo

$\Rightarrow$décodage facile

Types en Php

- Entiers
- Flottants
- Pas de booléens : faux=0
- Chaînes de caractère
- Tableaux indicés ou associatifs

Variables en Php

Variables sans déclaration, sans type

```
$x=1;
echo $x+32;

$x="toto";
echo $x;

define(BIDULE,35);
$x=BIDULE+28;
```

Typage

Attention : conversion implicite !

```
$r=1+’’les 8 pandas sont sympas’’;
```

⇒la chaîne est évaluée 0
⇒résultat 1

```
$r=1+’’8 pandas sont sympas’’;
```

⇒donne 9 (hum...)

Fonctions de la bibliothèque standard

Énormément de fonctions à tout faire

- ex: test du type

```
if (is_string($x))
  echo ‘‘x est une chaîne’’;
```

- ex: mise en minuscule

```
$x=’’L NE FAUT PAS CRIER’’
$x=strToLower($x);
```

$\Rightarrow$`$x` vaut `”il ne faut pas crier”`

Tableaux

- Indicés

```
$t=array(5,26,toto,132.2);
echo $t[0]; /* donne 5 */
echo $t[2]; /* donne toto */
```

- Associatifs (clé,donnée)

```
$t=array(Aymar=>Jean,
         Asstayé => Yves);
echo $t[Aymar]; /* donne Jean */
```

Chaînes de caractères

- Concaténation par “.”

```
$x=’’bonjour’’;
$x = $x . ’’ monsieur’’;
```

- Remplacement des variables dans la chaîne

```
$y = ’’J’ai dit $x !’’;
```

Modularité

Déclaration de fonctions

```
function bonjour($nom,$prenom){
  $resultat=''bonjour $prenom $nom'';
  return $resultat;
}
```

- `$nom`,`$prenom` et `$resultat` sont locales par défaut
- Passage par référence : `&$nom`
- Fonctions à nombre d'argument indéterminé

Importer du code Php

Exemple : fonction précédente dans `mesFonctions.php`

```
<?php
   require_once(''mesFonctions.php'');
   echo bonjour(''Buisson'',''George'');
?>
```

Objets

Exemple :

```
class Vehicule {
  var $reserve=0;
  function plein($litre){
    $this->reserve=$this->reserve+$litre;
  }
}

$vehicule=new Vehicule;
$vehicule->plein(20);
```

Objets : héritage

```
class Camion extends Vehicule {
  var $chargement;
  function charger($kilo){
    $this->chargement+=$kilo;
  }
}
$monCamion=new Camion;
$monCamion->plein(18);
```

Du bon usage de PHP

Quelques principes

- Décomposer le site en éléments d'affichage (menus, formulaires, publicité, zone d'information,...)
- À chaque élément, faire correspondre une fonction (ou un objet), avec éventuellement des paramètres
- Séparer les fonctions de mise en forme des fonctions de calcul (ex tableau HTML, contenu du tableau)
- Pas de valeur "en dur" : utiliser les constantes

Application

Une page avec

- En-tête
- Menu latéral
- Informations légales
- Page centrale...compliquée

Menu

```
function encre($texte,$lien){
  return "<A HREF=$lien>$texte</A>";
}
function creerMenu($tab){
  echo "<TABLE>";
  foreach($tab as $titre=>$lien){
    echo "<TR><TD>";
    echo encre($titre,$lien);
    echo "</TD></TR>";
  }
  echo "</TABLE>";
}
```

Menu

```
$menu=array(
   "Accueil"
      =>"http://localhost/Ex.html",
   "Google"
      =>"http://www.google.com",
   "Le Cnam"
      =>"http://www.cnam.fr");
```

En-tête

```
function enTete(){
  echo "<H1>Le gros portail</H1>";
  echo "<P>Sur ce site, vous trouverez ...heu...</P>";
}

define(AUTEUR,"David Gross-Amblard");
define(MAILAUTEUR,"dgram@cnam.fr");

function infoLegales(){
  echo "Copyright ".encre(AUTEUR,"mailto:".MAILAUTEUR);
}
```

Contenu principal

- Télécharger la page www.voila.fr
- Extraire les dépêches AFP (on connaît les balises utilisées)
- Reformuler avec notre présentation

Extraction d’information

```
function pagePrincipale($tab){
  echo "<H2>Les informations du jour</H2>";
  echo "<UL>";
  foreach($tab as $texte){
    echo "<LI>$texte";
  }
  echo "</UL>";
}
```

Extraction d’information

```
function extraitInfo($url){
  $src = implode("",file($url));
  preg_match_all(
"/<td class=\"b-actu\"[^>]*>[^<]+<\/td>[^<]+<td[^>]+><a[^>]*>(.*)<\/a>/m",
$src,$res);
  return $res[1];
}
```

Script final

```
<?php require_once("fonctions.php");?>
<TABLE BORDER>
<TR><TD><TD> <?php enTete(); ?> </TR>
<TR><TD><?php creerMenu($menu);?>
    <TD> <?php pagePrincipale(
                  extraitInfo("voila.html")
                ); ?>
</TR>
<TR><TD><TD><?php infoLegales(); ?>
<TR>
</TABLE>
```

démo

Conclusion

- Script côté serveur
- CGI : universel mais lourd
- PHP : un langage
  - Léger : intégré au serveur
  - Puissant : bibliothèque impressionnante
    - images
    - expressions régulière
    - formattage
    - BD, PDF, Flash, LDAP, ...
  - Peu typé (pb. pour grosses applications)