

Linux/apache/PHP vs Windows 2000/IIS/ASP

- > **1 - Introduction**
- > **2 - Architecture matériel**
- > **3 - Architecture logiciel**
- > **4 - Structure de la base de données de test**
- > **5 - Code du test de compilation**
- > **6 - Code du test de simulation d'affichage d'une liste de produits, d'articles, ou autres**
- > **7 - Code du test de simulation de livraison d'un bandeau ou d'affichage d'un produit**
- > **8 - Résultats du test de compilation**
- > **9 - Résultats du test de simulation d'affichage d'une liste de produits, d'articles, ou autres**
- > **10 - Résultats du test de simulation d'affichage d'un bandeau ou d'un produit**
- > **11 - Analyse des résultats**

1) Introduction

Nous avons réalisé trois tests au cours de ce benchmark : test de compilation, simulation d'affichage d'une liste de produits, d'articles, ou autres, simulation de livraison d'un bandeau ou d'affichage d'un produit

- Le test de compilation : permet de connaître la vitesse du compilateur associé au langage de script. ce test donne aussi la limite maximum en terme de capacité d'affichage pour ce langage de script.
- La simulation d'affichage d'une liste de produits, d'articles, ou autres et la simulation de livraison d'un bandeau ou d'affichage d'un produit : sont des situations couramment rencontrées sur les sites internet dynamique (moteur de recherche, forum, site e-commerce ...)

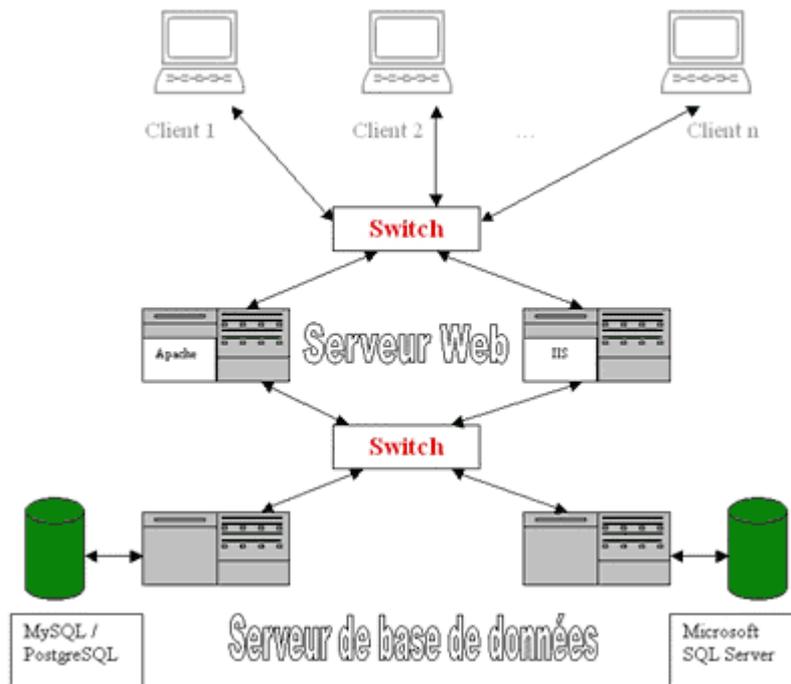
Ces différents tests permettront d'orienter le choix technologique pour une solution d'hébergement d'un site internet, intranet...

Ces tests sont réalisés avec un nombre de clients de 0 à 150 (connexions simultanées).

Le 15 Janvier 2002 par Philippe PETIT

2) Architecture matériel

Liste du matériel, configuration du réseau utilisée pour réaliser ce benchmark



Les serveurs web et de bases de données sont des serveurs Compaq Proliant 1850 R BI PII 400 Mhz avec 500Mo de RAM, 1 disque de 4 GO pour le système et 2 disques de 9 GO en miroir pour les données (site web). Les disques durs sont SCSI 10 000 tours, 2 cartes réseaux 100 Mb full duplex. Les deux réseaux (entre les clients et les serveurs web / entre les serveurs web et les serveurs de base de données) sont en 100 Mb.

3) Architecture logiciel

Liste des logiciels et des drivers utilisés pour réaliser ce benchmark

- Les divers clients sont des postes NT 4 ou Windows 2000 avec l'application Microsoft Web Application Stress version 1.1.293.1. Cette application permet de simuler l'accès simultané de plusieurs clients à un serveur web.
- Serveur web Apache : Système d'exploitation Linux Mandrake 8, Kernel 2.4.6, Apache 1.3.19, PHP 4.0.4pl1.
- Serveur web IIS : Système d'exploitation Windows 2000 serveur SP2 avec IIS 5 patché (code rouge...), MDAC 2.6 SP1, drivers MyODBC 2.50.39 et PgODBC 7.01.
- Serveur Base de Données PostgreSQL 7.03, MySQL 3.23.36 : Système d'exploitation Linux Mandrake 8, Kernel 2.4.6.
- Serveur Base de Données SQL server 2000 : Système d'exploitation Windows 2000 serveur SP2 avec MDAC 2.6 SP1.

4) Structure de la base de données de test

La structure est identique sur l'ensemble des serveurs de Base de Données (postgreSQL, MySQL, SQL server).

5000 enregistrements identiques (hors numéro automatique) :

idtest : (entier auto-incrément, servant d'index primaire)
testchar : (char 255)
testdate : (datetime)
testnum : (int)
testbool : (booléen)
testvarchar : (varchar 255)

Tous les champs peuvent être nul sauf idtest.

5) Code du test de compilation

Sources ASP et PHP permettant la réalisation du benchmark.

```
Sources PHP
<?PHP
echo "hello world";
?>
```

```
Sources ASP
<%
response.write("hello world")
%>
```

6) Code du test de simulation d'affichage d'une liste de produits, d'articles, ou autres

Test d'extraction des enregistrements 3001 à 3020 avec traitements et affichages de ceux-ci.
Ce test permet de simuler la création d'une liste.

PHP + MySQL 3.23.36

```
<HTML>
<BODY>
<?PHP
$dbconn = mysql_connect ("192.168.0.1", "root", "");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--> <br>";
$Affiche="";
$cpt=0;
$sql="SELECT * FROM test ORDER BY idtest LIMIT 3000, 20";
$sqlok=mysql_db_query ("testapache", $sql, $dbconn);
if ($sqlok) {
    while ($result=mysql_fetch_array ($sqlok, MYSQL_ASSOC)) {
        $cpt++;
        $templatetmp = $template;
        foreach ($result as $key => $value) {
            $templatetmp = eregi_replace("<!--".$key."-->",$value,$templatetmp);
        }
        $Affiche = $Affiche . "<li>Enregistrement " . $cpt . " : <br>" . $templatetmp;
    }
}
echo $Affiche;
mysql_close ($dbconn);
?>
</BODY>
</HTML>
```

PHP + connexions persistantes + MySQL 3.23.36

```
<HTML>
<BODY>
<?PHP
$dbconn = mysql_pconnect ("192.168.0.1", "root", "");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--> <br>";
$Affiche="";
$cpt=0;
$sql="SELECT * FROM test ORDER BY idtest LIMIT 3000, 20";
$sqlok=mysql_db_query ("testapache", $sql, $dbconn);
if ($sqlok) {
    while ($result=mysql_fetch_array ($sqlok, MYSQL_ASSOC)) {
        $cpt++;
        $templatetmp = $template;
        foreach ($result as $key => $value) {
            $templatetmp = eregi_replace("<!--".$key."-->",$value,$templatetmp);
        }
        $Affiche = $Affiche . "<li>Enregistrement " . $cpt . " : <br>" . $templatetmp;
    }
}
echo $Affiche;
mysql_close ($dbconn);
?>
</BODY>
</HTML>
```

Linux/apache/PHP vs Windows 2000/IIS/ASP

PHP + PostgreSQL 7.03

```
<HTML>
<BODY>
<?PHP
$dbconn = pg_Connect ("host=192.168.0.1 port=5432 dbname=testapache user=postgres password=");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--> <br>";
$Affiche="";
$spt=0;
$sql="SELECT * FROM test order by idtest LIMIT 20, 3000";
$dbpg_num_row=0;
$sqlok=pg_exec ($dbconn, $sql);
if ($sqlok) {
    while (pg_numrows($sqlok)>$dbpg_num_row) {
        $result=pg_fetch_array ($sqlok, $dbpg_num_row, PGSQL_ASSOC);
        $spt++;
        $templatetmp = $template;
        foreach ($result as $key => $value) {
            $templatetmp = eregi_replace("<!--" . strtoupper($key) . "-->", $value, $templatetmp);
        }
        $Affiche = $Affiche . "<li>Enregistrement " . $spt . " : <br>" . $templatetmp;
        $dbpg_num_row++;
    }
}
echo $Affiche;
pg_close ($dbconn);
?>
</BODY>
</HTML>
```

PHP + connexions persistantes + PostgreSQL 7.03

```
<HTML>
<BODY>
<?PHP
$dbconn = pg_pConnect ("host=192.168.0.1 port=5432 dbname=testapache user=postgres password=");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--> <br>";
$Affiche="";
$spt=0;
$sql="SELECT * FROM test order by idtest LIMIT 20, 3000";
$dbpg_num_row=0;
$sqlok=pg_exec ($dbconn, $sql);
if ($sqlok) {
    while (pg_numrows($sqlok)>$dbpg_num_row) {
        $result=pg_fetch_array ($sqlok, $dbpg_num_row, PGSQL_ASSOC);
        $spt++;
        $templatetmp = $template;
        foreach ($result as $key => $value) {
            $templatetmp = eregi_replace("<!--" . strtoupper($key) . "-->", $value, $templatetmp);
        }
        $Affiche = $Affiche . "<li>Enregistrement " . $spt . " : <br>" . $templatetmp;
        $dbpg_num_row++;
    }
}
echo $Affiche;
pg_close ($dbconn);
?>
</BODY>
</HTML>
```

Linux/apache/PHP vs Windows 2000/IIS/ASP

ASP + SQL Serveur 2000

```
<HTML>
<BODY>
<%
set MyCon = Server.CreateObject("ADODB.Connection")
MyCon.ConnectionTimeout = 1200
MyCon.ConnectionString = "DRIVER={SQL Server};DATABASE=testiis;SERVER=192.168.0.111;UID=root;PWD=root;"
MyCon.Open
template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - "
template=template & " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--><br>"
Affiche=""
cpt=0
sql="select * from (select top 20 * from (select top 3020 * from test order by idtest) as t1 order by idtest DESC) as t2 order by idtest"
Set rs = MyCon.Execute(sql)
while not rs.eof
  cpt = cpt + 1
  templatetmp = template
  for i=0 to rs.fields.count-1
    templatetmp = Replace(templatetmp,"<!--" & UCase(rs.fields(i).name) & "-->",rs.fields(i).value)
  next
  Affiche = Affiche & "<li>Enregistrement " & cpt & " : <br>" & templatetmp
  rs.MoveNext
Wend
rs.close
Set rs=Nothing
MyCon.Close
Set MyCon=Nothing
response.write Affiche
%>
</BODY>
</HTML>
```

La requête se trouvant dans ce script permet de simuler la fonction LIMIT de MySQL ou de PostgreSQL.

ASP + MyODBC 2.50.30 + MySQL 3.23.36

```
<HTML>
<BODY>
<%
set MyCon = Server.CreateObject("ADODB.Connection")
MyCon.ConnectionTimeout = 1200
MyCon.ConnectionString = "DRIVER={MySQL};DATABASE=testapache;SERVER=192.168.0.1;UID=root;PWD=;"
MyCon.Open
template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - "
template=template & " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--><br>"
Affiche=""
cpt=0
sql="SELECT * FROM test order by idtest LIMIT 3000, 20"
Set rs = MyCon.Execute(sql)
while not rs.eof
  cpt = cpt + 1
  templatetmp = template
  for i=0 to rs.fields.count-1
    templatetmp = Replace(templatetmp,"<!--" & UCase(rs.fields(i).name) & "-->",rs.fields(i).value)
  next
  Affiche = Affiche & "<li>Enregistrement " & cpt & " : <br>" & templatetmp
  rs.MoveNext
Wend
rs.close
```

Linux/apache/PHP vs Windows 2000/IIS/ASP

```
Set rs=Nothing
MyCon.Close
Set MyCon=Nothing
response.write Affiche
%>
</BODY>
</HTML>
```

ASP + PgODBC 7.01 + PostreSQL 7.03

```
<HTML>
<BODY>
<%
set MyCon = Server.CreateObject("ADODB.Connection")
MyCon.ConnectionTimeout = 1200
MyCon.ConnectionString =
"DRIVER={PostgreSQL};DATABASE=testapache;SERVER=192.168.0.1;UID=postgres;PWD="
MyCon.Open
template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - "
template=template & " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--><br>"
Affiche=""
cpt=0
sql="SELECT * FROM test order by idtest LIMIT 20, 3000"
Set rs = MyCon.Execute(sql)
while not rs.eof
  cpt = cpt + 1
  templatetmp = template
  for i=0 to rs.fields.count-1
    templatetmp = Replace(templatetmp,"<!--" & UCase(rs.fields(i).name) & "-->",rs.fields(i).value)
  next
  Affiche = Affiche & "<li>Enregistrement " & cpt & " : <br>" & templatetmp
  rs.MoveNext
Wend
rs.close
Set rs=Nothing
MyCon.Close
Set MyCon=Nothing
response.write Affiche
%>
</BODY>
</HTML>
```

7) Code du test de simulation de livraison d'un bandeau ou d'affichage d'un produit

Test d'extraction aléatoire d'un enregistrement avec traitement et affichage de celui-ci. Ce test permet de simuler l'affichage d'un bandeau, d'un produit...

PHP + MySQL 3.23.36

```
<HTML>
<BODY>
<?PHP
$dbconn = mysql_connect ("192.168.0.1", "root", "");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--> <br>";
$affiche="";
$spt=0;
srand(time());
$num=intval(rand()%5000)+1;
$sql="SELECT * FROM test where idtest=" . $num;
$sqlok=mysql_db_query ("testapache", $sql, $dbconn) ;
if ($sqlok) {
    while ($result=mysql_fetch_array ($sqlok, MYSQL_ASSOC)) {
        $spt++;
        $templatetmp=$template;
        foreach ($result as $key => $value) {
            $templatetmp = eregi_replace("<!--".$key."-->",$value,$templatetmp);
        }
        $affiche = $affiche . "<li>Enregistrement " . $spt . " : <br>" . $templatetmp;
    }
}
echo $affiche;
mysql_close ($dbconn);
?>
</BODY>
</HTML>
```

PHP + connexions persistantes + MySQL 3.23.36

```
<HTML>
<BODY>
<?PHP
$dbconn = mysql_pconnect ("192.168.0.1", "root", "");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--> <br>";
$affiche="";
$spt=0;
srand(time());
$num=intval(rand()%5000)+1;
$sql="SELECT * FROM test where idtest=" . $num;
$sqlok=mysql_db_query ("testapache", $sql, $dbconn) ;
if ($sqlok) {
    while ($result=mysql_fetch_array ($sqlok, MYSQL_ASSOC)) {
        $spt++;
        $templatetmp=$template;
        foreach ($result as $key => $value) {
            $templatetmp = eregi_replace("<!--".$key."-->",$value,$templatetmp);
        }
        $affiche = $affiche . "<li>Enregistrement " . $spt . " : <br>" . $templatetmp;
    }
}
echo $affiche;
mysql_close ($dbconn);
?>
</BODY>
</HTML>
```

PHP + PostgreSQL 7.03

```
<HTML>
<BODY>
<?PHP
$dbconn = pg_Connect ("host=192.168.0.1 port=5432 dbname=testapache user=postgres password=");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR-->
<br>";
$Affiche="";
$spt=0;
srand(time());
$num=intval(rand()%5000)+1;
$sql="SELECT * FROM test where idtest=" . $num;
$dbpg_num_row=0;
$sqlok=pg_exec ($dbconn, $sql);
if ($sqlok) {
  while (pg_numrows($sqlok)>$dbpg_num_row) {
    $result=pg_fetch_array ($sqlok, $dbpg_num_row, PGSQL_ASSOC);
    $spt++;
    $templatetmp = $template;
    foreach ($result as $key => $value) {
      $templatetmp = eregi_replace("<!--" . strtoupper($key) . "-->",$value,$templatetmp);
    }
    $Affiche = $Affiche . "<li>Enregistrement " . $spt . " : <br>" . $templatetmp;
    $dbpg_num_row++;
  }
}
echo $Affiche;
pg_close ($dbconn);
?>
</BODY>
</HTML>
```

PHP + connexions persistantes + PostgreSQL 7.03

```
<HTML>
<BODY>
<?PHP
$dbconn = pg_pConnect ("host=192.168.0.1 port=5432 dbname=testapache user=postgres password=");
$template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - ";
$template=$template . " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR-->
<br>";
$Affiche="";
$spt=0;
srand(time());
$num=intval(rand()%5000)+1;
$sql="SELECT * FROM test where idtest=" . $num;
$dbpg_num_row=0;
$sqlok=pg_exec ($dbconn, $sql);
if ($sqlok) {
  while (pg_numrows($sqlok)>$dbpg_num_row) {
    $result=pg_fetch_array ($sqlok, $dbpg_num_row, PGSQL_ASSOC);
    $spt++;
    $templatetmp = $template;
    foreach ($result as $key => $value) {
      $templatetmp = eregi_replace("<!--" . strtoupper($key) . "-->",$value,$templatetmp);
    }
    $Affiche = $Affiche . "<li>Enregistrement " . $spt . " : <br>" . $templatetmp;
    $dbpg_num_row++;
  }
}
echo $Affiche;
pg_close ($dbconn);
?>
```

Linux/apache/PHP vs Windows 2000/IIS/ASP

```
</BODY>
</HTML>
```

ASP + SQL Serveur 2000

```
<HTML>
<BODY>
<%
set MyCon = Server.CreateObject("ADODB.Connection")
MyCon.ConnectionTimeout = 1200
MyCon.ConnectionString = "DRIVER={SQL Server};DATABASE=testiis;SERVER=192.168.0.111;UID=root;PWD=root;"
MyCon.Open
template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - "
template=template & " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR-->
<br>"
Affiche=""
cpt=0
Randomize()
num = Int(5000 * Rnd)+1
sql="SELECT * FROM test where idtest=" & num
Set rs = MyCon.Execute(sql)
while not rs.eof
  cpt = cpt + 1
  templatetmp = template
  for i=0 to rs.fields.count-1
    templatetmp = Replace(templatetmp,"<!--" & UCase(rs.fields(i).name) & "-->",rs.fields(i).value)
  next
  Affiche = Affiche & "<li>Enregistrement " & cpt & " : <br>" & templatetmp
  rs.MoveNext
Wend
rs.close
Set rs=Nothing
MyCon.Close
Set MyCon=Nothing
response.write Affiche
%>
</BODY>
</HTML>
```

ASP + MyODBC 2.50.30 + MySQL 3.23.36

```
<HTML>
<BODY>
<%
set MyCon = Server.CreateObject("ADODB.Connection")
MyCon.ConnectionTimeout = 1200
MyCon.ConnectionString = "DRIVER={MySQL};DATABASE=testapache;SERVER=192.168.0.1;UID=root;PWD=;"
MyCon.Open
template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - "
template=template & " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR-->
<br>"
Affiche=""
cpt=0
Randomize()
num = Int(5000 * Rnd)+1
sql="SELECT * FROM test where idtest=" & num
Set rs = MyCon.Execute(sql)
while not rs.eof
  cpt = cpt + 1
  templatetmp = template
  for i=0 to rs.fields.count-1
    templatetmp = Replace(templatetmp,"<!--" & UCase(rs.fields(i).name) & "-->",rs.fields(i).value)
  next
  Affiche = Affiche & "<li>Enregistrement " & cpt & " : <br>" & templatetmp
  rs.MoveNext
%>
```

Linux/apache/PHP vs Windows 2000/IIS/ASP

```
Wend
rs.close
Set rs=Nothing
MyCon.Close
Set MyCon=Nothing
response.write Affiche
%>
</BODY>
</HTML>
```

ASP + PgODBC 7.01 + PostreSQL 7.03

```
<HTML>
<BODY>
<%
set MyCon = Server.CreateObject("ADODB.Connection")
MyCon.ConnectionTimeout = 1200
MyCon.ConnectionString =
"DRIVER={PostgreSQL};DATABASE=testapache;SERVER=192.168.0.1;UID=postgres;PWD="
MyCon.Open
template="idtest : <!--IDTEST--> - testchar : <!--TESTCHAR--> - testdate : <!--TESTDATE--> - "
template=template & " testnum : <!--TESTNUM--> - testbool : <!--TESTBOOL--> - testvarchar : <!--TESTVARCHAR--><br>"
Affiche=""
cpt=0
Randomize()
num = Int(5000 * Rnd)+1
sql="SELECT * FROM test where idtest=" & num
Set rs = MyCon.Execute(sql)
while not rs.eof
  cpt = cpt + 1
  templatetmp = template
  for i=0 to rs.fields.count-1
    templatetmp = Replace(templatetmp,"<!--" & UCase(rs.fields(i).name) & "-->",rs.fields(i).value)
  next
  Affiche = Affiche & "<li>Enregistrement " & cpt & " : <br>" & templatetmp
  rs.MoveNext
Wend
rs.close
Set rs=Nothing
MyCon.Close
Set MyCon=Nothing
response.write Affiche
%>
</BODY>
</HTML>
```

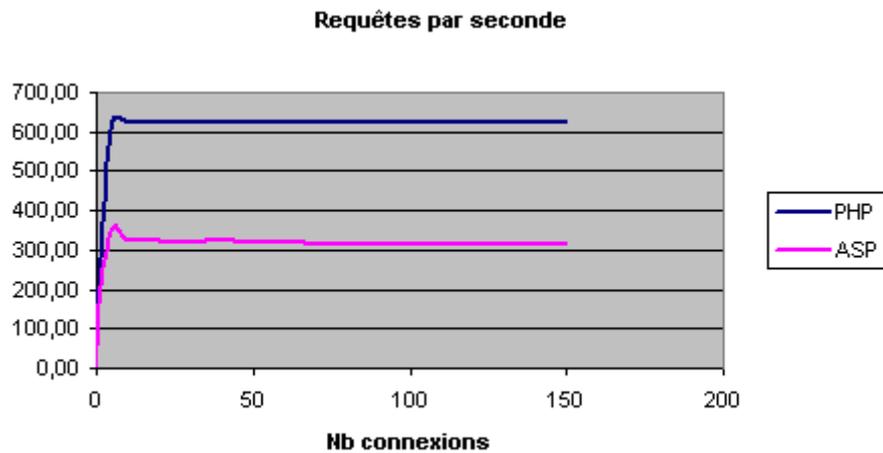
8) Résultats du test de compilation

Le test de compilation permet de connaître la vitesse du compilateur associé au langage de script. ce test donne aussi la limite maximum en terme de capacité d'affichage pour ce langage de script.

Requêtes par seconde

Nb Connexions PHP ASP

Requêtes par seconde		
Nb Connexions	PHP	ASP
0	0,00	0,00
1	276,88	210,32
5	627,05	355,55
10	625,27	325,90
20	625,72	327,02
30	626,27	322,02
40	625,92	323,52
50	626,03	320,87
100	623,58	316,45
150	622,20	316,90



Linux/apache/PHP vs Windows 2000/IIS/ASP

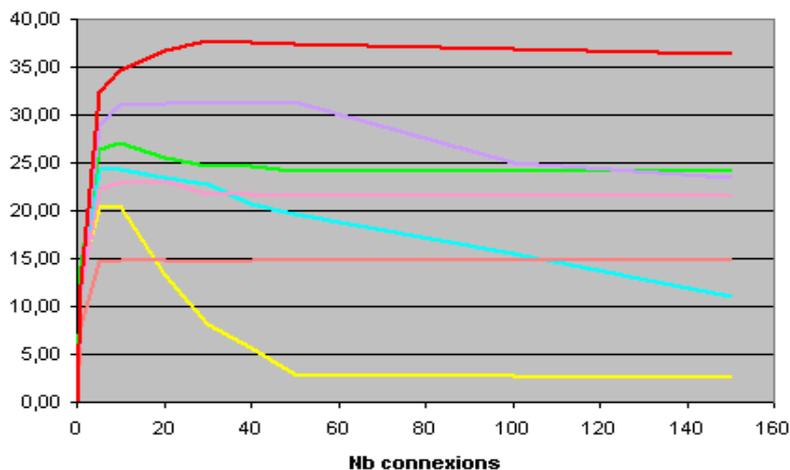
9) Résultats du test de simulation d'affichage d'une liste de produits, d'articles, ou autres

La simulation d'affichage d'une liste de produits, d'articles, ou autres est une situation couramment rencontrée sur les sites internet dynamique (moteur de recherche, forum, site e-commerce ...)

	PHP + MySQL 3.23.36
	PHP + connexions persistantes + MySQL 3.23.36
	PHP + PostgreSQL 7.0.3
	PHP + connexions persistantes + PostgreSQL 7.0.3
	ASP + SQL Server 2000
	ASP + MyODBC 2.50.39 + MySQL 3.23.36
	ASP + pgODBC 7.01 + PostgreSQL 7.0.3

Requêtes par seconde							
	Linux Mandrake 8.0 Apache				Windows 2000 IIS5		
Nb Connexions							
0	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	13,12	13,00	7,80	11,55	14,08	11,75	12,18
5	20,32	24,28	14,72	22,27	26,42	29,03	32,45
10	20,57	24,23	14,75	22,92	27,13	30,97	34,63
20	13,27	23,52	14,78	23,03	25,60	31,25	36,72
30	8,12	22,77	14,70	21,73	24,65	31,17	37,78
40	5,68	20,72	14,77	21,72	24,63	31,20	37,58
50	2,88	19,73	14,78	21,60	24,20	31,40	37,42
100	2,78	15,53	14,75	21,55	24,20	24,95	36,88
150	2,67	11,02	14,77	21,47	24,20	23,43	36,45

Requêtes par seconde



Linux/apache/PHP vs Windows 2000/IIS/ASP

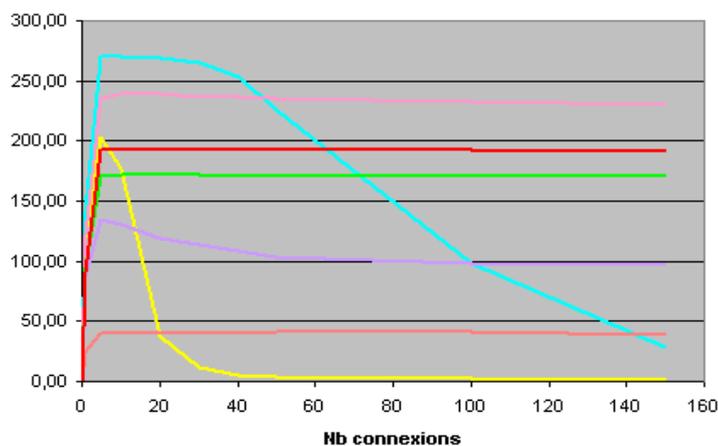
10) Résultats du test de simulation d'affichage d'un bandeau ou d'un produit

La livraison d'un bandeau ou d'affichage d'un produit est une situation couramment rencontrée sur les sites internet dynamique (moteur de recherche, régie publicitaire, forum, site e-commerce ...)

	PHP + MySQL 3.23.36
	PHP + connexions persistantes + MySQL 3.23.36
	PHP + PostgreSQL 7.0.3
	PHP + connexions persistantes + PostgreSQL 7.0.3
	ASP + SQL Server 2000
	ASP + MyODBC 2.50.39 + MySQL 3.23.36
	ASP + pgODBC 7.01 + PostgreSQL 7.0.3

Requêtes par seconde							
	Linux Mandrake 8.0 Apache				Windows 2000 IIS5		
Nb Connexions							
0	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	118,68	141,42	24,32	121,88	93,82	86,07	90,77
5	203,63	270,90	40,47	235,70	171,02	135,10	192,60
10	177,35	270,60	40,62	239,70	171,65	130,37	193,18
20	37,57	268,87	40,60	238,88	171,58	119,32	193,35
30	11,20	264,85	40,63	236,73	171,52	113,88	193,05
40	4,53	253,60	40,52	236,58	171,20	109,10	192,22
50	4,23	226,00	40,87	235,28	171,13	103,87	192,78
100	2,42	98,55	40,97	233,15	171,65	97,93	192,30
150	1,55	29,05	39,03	230,68	171,05	97,57	191,88

Requêtes par seconde



12) Analyse des résultats

Les chiffres indiqués sont donnés pour faire une comparaison entre les différentes solutions. Ces valeurs peuvent changer suivant la configuration matérielle et la complexité des scripts utilisés.

Toutefois, nous avons pris les précautions nécessaires pour que la hiérarchie des performances se dégageant du benchmark soit sensiblement la même si on utilise un autre matériel ou d'autres scripts.

Ces conclusions ne sont valables que dans la configuration préconisée c'est à dire un serveur web frontal associé à un serveur SGDB. Etant donné que les performances sont tributaires de la configuration matérielle, les chiffres ci dessous sont donnés à titre indicatif.

Il est admis qu'une pointe de trafic d'un site dépasse rarement le double du trafic moyen. Dans notre analyse, pour se ménager un peu de sécurité, nous allons prendre un coefficient de 2,5.

En regardant la première courbe du benchmark (compilation sans accès base de données), on voit que les deux langages restent sensiblement équivalents jusqu'à 300 requêtes/s. Au-delà, le PHP est plus performant. En prenant un coefficient de 2,5 pour les pics de trafic, les 300 requêtes/s du PHP correspondent à 10,37 Millions de pages vues par jour. Donc, en ne prenant en compte que l'aspect compilation, jusqu'à 10 millions de pages vues par jour ASP et PHP sont équivalents. Au-delà, PHP prend le meilleur.

Maintenant, prenons en compte la connexion base de données. Nous avons fait 2 tests correspondant chacun à un type d'affichage :

- Simulation d'affichage d'une liste de produits, d'articles, ou autres
- Simulation de livraison d'un bandeau ou d'affichage d'un produit

Dans un premier temps, on peut remarquer que les performances du couple PHP - MySQL décroissent dangereusement avec le nombre de connexions simultanées. Donc, il est préférable de ne pas retenir cette solution pour des sites à fort trafic.

Ensuite le choix de la solution dépend du type de site à héberger. Pour les sites à contenu, les catalogues en ligne ou les moteurs de recherche, la majorité des pages à afficher sont des listes (correspond à la première simulation). Par conséquent, dans ces cas, la meilleure solution est le couple ASP - PostgreSQL : 37 requêtes/s, soit 1,27 Millions de pages vues par jour. Pour ceux qui souhaitent garder le même système d'exploitation sur le serveur web et le serveur de base de données, ASP - SQL Server 2000 reste une solution intéressante : 25 requêtes/s, soit 864 000 pages vues par jour.

Par contre, si le site à héberger est prévu pour de la livraison de bandeaux ou tout autre service consistant à afficher un élément d'une table de la base de données, le dernier test du benchmark révèle que la solution PHP - PostgreSQL est la meilleure : 238 requêtes/s, soit 8,22 Millions de pages vues par jour. Le couple ASP - PostgreSQL a des performances encore intéressantes : 193 requêtes/s, soit 6,67 Millions de pages mais l'accès à des champs de type MEMO (BLOB) pour le couple ASP - PostgreSQL est plus compliqué que pour le couple ASP - SQL Server..

Philippe PETIT