



Πτυχιακή Εργασία

Ανάπτυξη Σελίδας Στατιστικών Για Τον Web Server Του Τμήματος

Βαβάλας Απόστολος

Ιούνιος 2009

Α.Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

A.T.E.I.Θ.
Τμήμα Πληροφορικής

Θέμα Πτυχιακής Εργασίας :
Ανάπτυξη Σελίδας Στατιστικών για τον Web
Server του Τμήματος

Βαβάλας Απόστολος 03/2383
apvan@it.teithe.gr



Περιεχόμενα

| | | |
|----------|------------------------------------|-----------|
| 1 | ΠΡΟΛΟΓΟΣ | 7 |
| 2 | ΕΙΣΑΓΩΓΗ | 9 |
| 2.1 | ΣΚΟΠΟΣ | 9 |
| 2.2 | ΑΠΑΙΤΗΣΕΙΣ | 9 |
| 2.3 | SERVER ΑΕΤΟΣ | 10 |
| 2.3.1 | Εγκατεστημένα Πακέτα | 12 |
| 2.4 | ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ | 12 |
| 2.4.1 | Python | 12 |
| 2.4.2 | PHP | 13 |
| 2.4.2.1 | GD2 | 13 |
| 2.4.3 | Smarty Template Engine | 14 |
| 2.4.4 | Graphite | 15 |
| 2.4.5 | CSS | 17 |
| 3 | ΥΠΑΡΧΟΝΤΑ ΕΡΓΑΛΕΙΑ | 19 |
| 3.1 | AWSTATS | 19 |
| 3.2 | ANALOG | 20 |
| 3.3 | WEBALIZER | 20 |
| 3.4 | ΜΕΤΡΗΤΕΣ (COUNTERS) | 21 |
| 4 | ΔΟΜΗ / ΑΝΑΛΥΣΗ | 23 |
| 4.1 | ΓΛΩΣΣΕΣ ΥΛΟΠΟΙΗΣΗΣ | 23 |
| 4.1.1 | Parser | 23 |
| 4.1.2 | Βάση Δεδομένων | 24 |
| 4.1.3 | Web Interface | 24 |
| 4.2 | ΣΧΗΜΑ ΒΑΣΗΣ | 24 |
| 4.2.1 | Πίνακες Βάσης | 26 |
| 4.2.2 | SQL κώδικας δημιουργίας των tables | 28 |
| 4.2.3 | Σχεδιάγραμμα ER της Βάσης | 31 |
| 5 | ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ | 33 |
| 5.1 | LAUNCHER.SH | 33 |
| 5.1.1 | Κώδικας του launcher.sh | 33 |

| | | |
|----------|--|-----------|
| 5.2 | PARSER.PY | 33 |
| 5.3 | WEB INTERFACE | 39 |
| 5.3.1 | Προγραμματιστικό μέρος web interface | 40 |
| 5.3.2 | Γραφήματα του graphs.php | 43 |
| 5.3.3 | Counter | 47 |
| 6 | ΕΓΚΑΤΑΣΤΑΣΗ | 51 |
| 6.1 | ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ | 51 |
| 6.2 | ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ | 51 |
| 6.2.1 | Εγκατάσταση counter σε ιστοσελίδα | 53 |
| 7 | ΧΡΗΣΗ | 55 |
| 7.1 | SHELL ΕΝΤΟΛΕΣ | 55 |
| 7.2 | ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ | 56 |
| 8 | ΣΧΟΛΙΑ / ΠΑΡΑΤΗΡΗΣΕΙΣ | 63 |
| 9 | ΕΠΙΛΟΓΟΣ | 65 |

1 ΠΡΟΛΟΓΟΣ

Η πτυχιακή εργασία που αναπτύσσεται σε αυτό το κείμενο έχει θέμα “Ανάπτυξη Σελίδας Στατιστικών για τον Web Server του τμήματος”. Πρόκειται για την υλοποίηση ενός εργαλείου που συλλέγει στοιχεία για τις επισκέψεις στον server aetos από τα access logs της HTTP υπηρεσίας και τα παρουσιάζει μέσω ενός web interface. Οι απαιτούμενες γνώσεις για αυτήν την εργασία προέρχονται από διάφορους κλάδους της επιστήμης της πληροφορικής όπως ο προγραμματισμός, οι βάσεις δεδομένων και τα λειτουργικά συστήματα.

Στο κεφάλαιο Εισαγωγή αναφέρονται τα χαρακτηριστικά του web server του τμήματος και παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Στο επόμενο κεφάλαιο γίνεται αναφορά σε υπάρχοντα εργαλεία που αναλούν web logs, στα χαρακτηριστικά τους και παρουσιάζονται οι διαφορές τους. Το κεφάλαιο τέσσερα είναι αφιερωμένο στην παρουσίαση της δομής της υπηρεσίας καθώς και στους λόγους επιλογής των εργαλείων υλοποίησης για κάθε επιμέρους δομικό στοιχείο της. Στη συνέχεια αναλύεται ο κώδικας της εφαρμογής και ο τρόπος λειτουργίας του. Το έκτο κεφάλαιο αναφέρεται στις απαιτήσεις τους συστήματος και δίνονται οδηγίες για την εγκατάσταση του εργαλείου. Τέλος δίνονται αναλυτικές οδηγίες χρήσης, παρουσιάζονται στοιχεία από τη λειτουργία της εφαρμογής στο server του τμήματος καθώς και φωτογραφίες από την ιστοσελίδα της υπηρεσίας.

Η web based εφαρμογή αναπτύχθηκε σε περιβάλλον Unix χρησιμοποιώντας ελεύθερο λογισμικό και ονομάστηκε webstats. Η ηλεκτρονική διεύθυνση της ιστοσελίδας της υπηρεσίας είναι :

- <http://www.it.teithe.gr/~apvav/webstats>

2 ΕΙΣΑΓΩΓΗ

2.1 ΣΚΟΠΟΣ

Ο σκοπός της εφαρμογής είναι η ύπαρξη ενός εργαλείου που παρουσιάζει στατιστικά μεγέθη σχετικά με τις επισκέψεις στους ιστότοπους των λογαριασμών του server του τμήματος. Το εργαλείο είναι χτισμένο επάνω στις απαιτήσεις και στις ιδιαιτερότητες του aetos, η σημαντικότερη από αυτές είναι ότι φιλοξενεί πολλούς ιστότοπους λόγω των πολλών λογαριασμών. Περισσότερα σχετικά με τον server του τμήματος παρουσιάζονται στην υποενότητα 2.3.

Σκοπός της εφαρμογής δεν αποτελεί μόνο η εμφάνιση στοιχείων για τις επισκέψεις στην κάθε ιστοσελίδα αλλά και η ενημέρωση στο τι είδους ιστότοποι παρουσιάζουν το μεγαλύτερο ενδιαφέρον. Για αυτόν το λόγο οι χρήστες του aetos κατηγοριοποιήθηκαν ως προς την ιδιότητα τους στο τμήμα, δηλαδή σε προσωπικό και φοιτητές, έτσι δώθηκε η δυνατότητα υπολογισμού στατιστικών ειδικά για αυτές τις κατηγορίες. Ενδιαφέρον θεωρήθηκε και ο υπολογισμός διαφόρων άλλων στοιχείων όπως οι χρήστες και τα χρονικά διαστήματα με τις περισσότερες επισκέψεις.

2.2 ΑΠΑΙΤΗΣΕΙΣ

Οι απαιτήσεις μας από αυτό το εργαλείο δεν είναι άλλες από την ενημέρωση μας για την web κίνηση στο server με στατιστικά μεγέθη και σχεδιαγράμματα για την ασφαλή εξαγωγή συμπερασμάτων σχετικά με αυτή. Σημαντικό είναι τα στοιχεία να εμφανίζονται και για κάθε χρήστη του server ξεχωριστά. Τα στοιχεία αυτά είναι ο αριθμός των επισκέψεων και ο όγκος των δεδομένων που έχουν μεταφερθεί. Μία ακόμα απαίτηση είναι η παρουσίαση αυτών των μεγεθών, όχι μόνο ως προς το σύνολο τους αλλά και για ξεχωριστά χρονικά διαστήματα.

Αυτά τα διαστήματα είναι :

- Η τρέχουσα ώρα
- Η τρέχουσα ημέρα
- Ο τρέχων μήνας
- Οι τελευταίοι τρεις μήνες
- Οι τελευταίοι έξι μήνες
- Το τρέχων έτος

- Τα τελευταία δύο έτη
- Συνολικά

Όπως αναφέρθηκε παραπάνω οι λογαριασμοί απαιτείται να χωρίζονται σε κατηγορίες προσωπικού και φοιτητών. Πέραν αυτών δημιουργήθηκε μία ακόμα κατηγορία για την καλυψη μελλοντικών αναγκών με όνομα other. Αυτή η απαίτηση οφειλόταν στο να υπάρχει η δυνατότητα υπολογισμού της κατανομής των επισκέψεων ανά κατηγορία και των ποσοστών τους επί τοις εκατό.

Ένας επιπλέον λόγος αυτής της ομαδοποίησης είναι η ενημέρωση για την web κίνηση σε μία μεγαλύτερη κλίμακα και την παρουσίαση στοιχείων αρθροιστικά για τις ιστοσελίδες του προσωπικού και των φοιτητών. Έτσι μας δίνεται η δυνατότητα να γνωρίζουμε πως μοιράζεται η κίνηση στον server και ποια είναι η κατανομή των επισκέψεων ανάμεσα στην κεντρική σελίδα του τμήματος, δηλαδή τις ιστοσελίδες στο root directory του apache, τους ιστότοπους του προσωπικού και των φοιτητών.

Ακόμη μία απαίτηση από αυτήν την υπηρεσία είναι η ενημέρωση στατιστικών μεγεθών όπως μέσοι όροι αλλά και η αναλυτική παρουσίαση των επισκέψεων για χρονικά διαστήματα όπως οι τελευταίες ώρες, μέρες, μήνες και έτη. Οι κατανομές που υπολογίζονται είναι η κατανομή ανά κατηγορία όπου αναφερθήκαμε στην προηγούμενη παράγραφο και η κατανομή ανά ώρα, δηλαδή πως κατανέμεται η επισκεψιμότητα κατά τη διάρκεια μιας ημέρας. Το να γνωρίζουμε το ποιες ώρες γίνονται οι περισσότερες επισκέψεις και ποιές όχι είναι ένα πολύ ενδιαφέρον στατιστικό στοιχείο.

Για την εμφάνιση μεγεθών όπως οι κατανομές στις οποίες αναφερθήκαμε αλλά και η αναπαράσταση της κίνησης για ένα χρονικό διάστημα θεωρήθηκε πως η χρήση γραφημάτων θα ήταν ιδιαίτερα χρήσιμη για τη κατανόηση τους. Σχεδιαγράμματα παρουσιάζουν τη διακύμανση της επισκεψιμότητας για τα χρονικά διαστήματα των τελευταίων ωρών, ημερών, μηνων και ετών. Επίσης και για τους μήνες του κάθε έτους ξεχωριστά.

2.3 SERVER AETOS

Όπως προαναφέρθηκε το εργαλείο webstats δημιουργήθηκε ειδικά για τον aetos, τον server του τμήματος. Αυτό σημαίνει πως χτίστηκε σύμφωνα με τα χαρακτηριστικά και τις ρυθμίσεις του aetos, για αυτό θεωρήθηκε σκόπιμο να γίνει ειδική αναφορά σε αυτόν και τα χαρακτηριστικά του.

Βασικά χαρακτηριστικά του server του τμήματος :

- Το λειτουργικό σύστημα είναι Debian GNU/Linux 5.0
- Ο http υπηρεσία που τρέχει είναι ο Apache 2.2.9
- Ο web server έχει ενεργοποιημένη την usredir επιλογή, έτσι ο κάθε χρήστης μπορεί να ανεβάσει ιστοσελίδα. Η εντολή για την ενεργοποίηση αυτής της ρύθμισης είναι :

```
a2enmod userdir
```

- Ο web server ακούει σε δύο domain names. Αυτά είναι :
 1. `http://www.it.teithe.gr`
 2. `http://aetos.it.teithe.gr`
- Η ηλεκτρονική διεύθυνση των ιστοσελίδων των χρηστών είναι το url του aetos ακολουθούμενο από τον χαρακτήρα `~` και το όνομα του λογαριασμού. Παραδείγματος χάριν για τον χρήστη `arvan` :
 1. `http://www.it.teithe.gr/~arvan` ή
 2. `http://aetos.it.teithe.gr/~arvan`
- Ένα ακόμα feature που είναι ενεργοποιημένο στον Apache στον aetos είναι το `suEXEC`. Αυτή η λειτουργία επιτρέπει στους χρήστες του web server να τρέχουν CGI προγράμματα κάτω από διαφορετικό ID από αυτό της web υπηρεσίας, έτσι μειώνεται το ρίσκο σε θέματα ασφάλειας.
- Τα access logs του Apache αποθηκεύονται στα παρακάτω δύο αρχεία, ένα για κάθε domain name.
 1. `/var/log/apache2/aetos-access.log`
 2. `/var/log/apache2/www-access.log`
- Όσον αφορά τα access logs θα πρέπει να αναφερθεί το format τους πάνω στο οποίο υλοποιήθηκε το parsing. Το format με το οποίο αποθηκεύονται τα logs στον aetos είναι το `combined`. Το `webstats` υποστηρίζει επιπλέον το `common` format μόνο που σε αυτήν την περίπτωση δε λειτουργεί το φιλτράρισμα στα logs που προέρχονται από robots και spiders. Η `combined` μορφή είναι η `common` με την προσθήκη δύο ακόμα πεδίων, του `referer` και του `user-agent`. Η μορφή των δύο αυτών format είναι :
 1. `LogFormat "%h %l %u %t \"%r\" %>s %b" common`
 Παράδειγμα :
 - `127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326`
 2. `LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""`
`combined`
 Παράδειγμα :
 - `127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html" "Mozilla/4.08 [en] (Win98; I;Nav)"`

- Το webstats συνδέεται στην PostgreSQL τοπικά. Για να επιτραπεί μια δικτυακή σύνδεση στην PostgreSQL στον aetos θα πρέπει να γίνει από IP διεύθυνση που ανήκει στο δίκτυο του Τ.Ε.Ι. Θεσσαλονίκης.

2.3.1 Εγκατεστημένα Πακέτα

Παρακάτω ακολουθεί η λίστα των εγκατεστημένων πακέτων στον aetos που χρησιμοποιούνται από το webstats.

- python
- postgresql
- python-pgsql
- php5-pgsql
- php5
- php5-gd
- smarty
- logtail

Για την εγκατάσταση των παραπάνω πακέτων στα Debian απαιτείται η εντολή :

```
apt-get install package-name
```

2.4 ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

2.4.1 Python

Η python είναι μία high-level γλώσσα προγραμματισμού και αναπτύσσεται ως ανοιχτό λογισμικό. Έχει δοθεί ιδιαίτερη έμφαση στο συντακτικό της ώστε να είναι ευανάγνωστο και απλό. Υποστηρίζει πολλά προγραμματιστικά στυλ και εκτελείτε με την κλήση interpreter. Συχνά χρησιμοποιείται ως scripting language σε web applications. Βρίσκεται σε όλες τις Linux διανομές, NetBSD και OpenBSD ως βασικό component.

- Ακολουθεί ένα παράδειγμα κώδικα που δείχνει πόσο λιτό είναι το συντακτικό της python :

```
1 #!/usr/bin/python
2
3 def function1(a,b):
4     x = a + b;
5     return x;
6
7 print "Python_Example";
8 x = function1(4,3);
9 print x;
```

2.4.2 PHP

Η PHP είναι μία scripting language που δημιουργήθηκε για να παράγει ιστοσελίδες με δυναμικό περιεχόμενο, στην συνέχεια εξελίχτηκε σαν μια γλώσσα γενικότερων σκοπών. Χρησιμοποιείται κυρίως για side server programming και web applications καθώς μπορεί να ενσωματώσει HTML κώδικα. Είναι ελεύθερο λογισμικό και διατίθεται σχεδόν για όλα τα λειτουργικά συστήματα.

- Παράδειγμα :

```
<HTML>
  <HEAD>
    <TITLE> PHP </TITLE>
  </HEAD>
  <BODY>
    <?php echo "<p>Dynamic output.</p>"; ?>
    <p>Static output.</p>
  </BODY>
</HTML>
```

2.4.2.1 GD2

Μία από της standard βιβλιοθήκες της PHP είναι και η GD 2 που είναι εγκατεστημένη στον aetos. Αυτή η βιβλιοθήκη δίνει την δυνατότητα δημιουργίας εικόνων δυναμικά. Μπορεί να παράγει αρχεία όπως gif, jpg και png.

- Παράδειγμα :

```
<?php
// Create a 150*20 image
$im = imagecreate(150, 20);
// blue background and white text
```

```

$bg = imagecolorallocate($im, 255, 255, 255);
$textcolor = imagecolorallocate($im, 0, 0, 255);
imagestring($im, 5, 0, 0, 'Created by PHP', $textcolor);
// Output the image
header('Content-type: image/png');
imagepng($im);
imagedestroy($im);
?>

```

Το παραπάνω παράδειγμα θα δημιουργήσει αυτήν την εικόνα :



Figure 2.1: Παράδειγμα GD

2.4.3 Smarty Template Engine

Το Smarty είναι ένα Web Template σύστημα. Είναι γραμμένο σε PHP και δημιουργήθηκε για να διαχωρίζει το προγραμματιστικό μέρος από αυτό της εμφάνισης. Με αυτόν τον τρόπο αποφεύγεται ο συνδυασμός php κώδικα με html tags.

Υποστηρίζει δομές ελέγχου και επανάληψης, δήλωση μεταβλητών, δημιουργία συναρτήσεων και δίνει την δυνατότητα σε υψηλού επιπέδου template programming.

- Παράδειγμα PHP κώδικα που χρησιμοποιεί το Smarty :

```

<?php
//put full path to Smarty.class.php
require('/usr/share/php/smarty/libs/Smarty.class.php');
$smarty = new Smarty();
$smarty->template_dir = "/var/www/smarty/templates";
$smarty->compile_dir = "/var/www/smarty/templates_c";
$smarty->cache_dir = "/var/www/smarty/cache";
$smarty->config_dir = "/var/www/smarty/configs";

$x="php";

$smarty->assign('x', $x);
$smarty->display("mytemplate.tpl");
?>

```

- Παράδειγμα Smarty κώδικα από template του webstats :

```
<title>WebStats | Apotelesmata
    {if $group==NULL }
        {if $uname eq ':main'} Kentriki Selida
        {else}
            {if $usrdir != NULL} {$uname} {/if}
        {/if}
    {else}
        {if $group eq 'all'} Server
        {else} {$group}
        {/if}
    {/if}
</title>
```

2.4.4 Graphpите

Η Graphpите είναι μία βιβλιοθήκη ανοιχτού κώδικα γραμμένη σε PHP. Χρησιμοποιείται τη δημιουργία γραφημάτων διαφόρων τύπων τα οποία εξάγονται σε αρχεία τύπου jpg και png. Λειτουργεί με την PHP 4 και 5 ενώ απαιτείται και η βιβλιοθήκη GD 2, η χρήση της GD 1.x έχει σαν αποτέλεσμα την μείωση των δυνατοτήτων της. Το 2004 διατέθηκε η πρώτη stable έκδοση και το ίδιο έτος ενσωματώθηκε στην PEAR με το όνομα Image_Graph. Στο εργαλείο webstats χρησιμοποιείτε η 1.2, η πιο πρόσφατη έκδοση.

- Παράδειγμα κώδικα :

```
1 <?php
2     include("Graph.php");
3     // create the graph
4     $Graph =& new Image_Graph(400, 300);
5     // add a TrueType font
6     $Arial =& $Graph->addFont(new Image_Graph_Font_TTF("
7         arial.ttf"));
8     // set the font size to 12 pixels
9     $Arial->setSize(12);
10
11     // create the plotarea
12     $Graph->add(
13         new Image_Graph_Layout_Vertical(
14             new Image_Graph_Title("GraPHPite", $Arial),
15             $Plotarea = new Image_Graph_Plotarea(),
16             5)
```

```
16     );
17
18     // create the dataset
19     $Dataset =& new Image_Graph_Dataset_Trivial();
20     $Dataset->addPoint(1, 574);
21     $Dataset->addPoint(2, 239);
22     $Dataset->addPoint(3, 471);
23
24     $GridY =& $Plotarea->addGridY(new Image_Graph_Grid_Bars
25         ());
26     $GridY->setFillStyle(new Image_Graph_Fill_Gradient(
27         IMAGE_GRAPH_GRAD_VERTICAL, IMAGE_GRAPH_WHITE,
28         IMAGE_GRAPH_LIGHTGRAY, 100));
29
30     $FillArray->add(new Image_Graph_Fill_Gradient(
31         IMAGE_GRAPH_GRAD_RADIAL, IMAGE_GRAPH_BLACK,
32         IMAGE_GRAPH_WHITE, 200));
33     $FillArray->add(new Image_Graph_Fill_Gradient(
34         IMAGE_GRAPH_GRAD_RADIAL, IMAGE_GRAPH_BLUE,
35         IMAGE_GRAPH_WHITE, 200));
36     $FillArray->add(new Image_Graph_Fill_Gradient(
37         IMAGE_GRAPH_GRAD_RADIAL, IMAGE_GRAPH_ORANGE,
38         IMAGE_GRAPH_WHITE, 200));
39
40     $AxisX =& $Plotarea->getAxis(IMAGE_GRAPH_AXIS_X);
41     $AxisX->setDataPreprocessor(
42         new Image_Graph_DataPreprocessor_Array(
43             array(1=>"A", 2=>"B", 3=>"C", 4=>"")
44         )
45     );
46
47     $AxisX->setLabelInterval(1);
48     $AxisY =& $Plotarea->getAxis(IMAGE_GRAPH_AXIS_Y);
49
50     // output the Graph
51     $Graph->done();
52 ?>
```

Ο κώδικας παράγει το παρακάτω γράφημα :

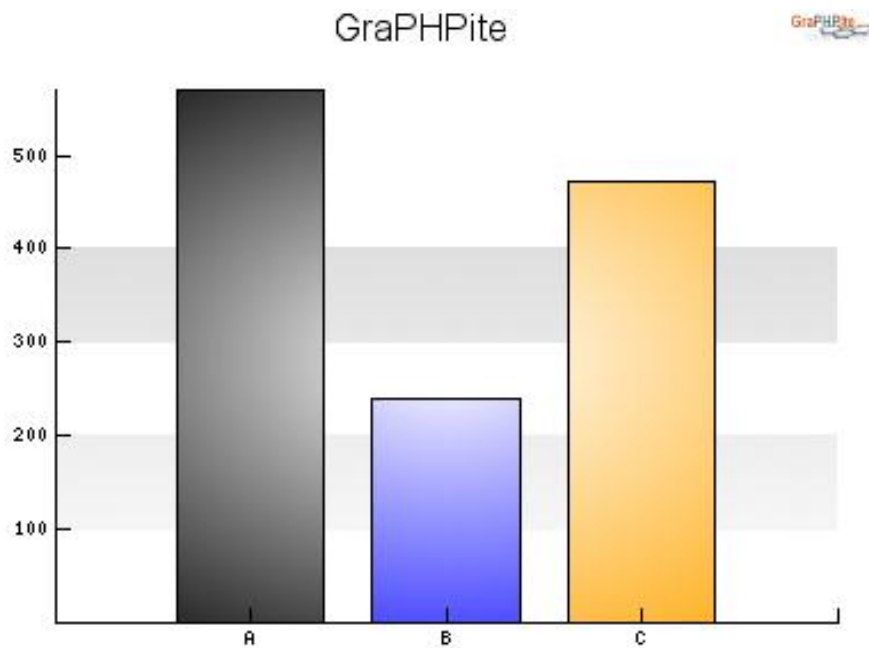


Figure 2.2: Graphite Γράφημα

2.4.5 CSS

Cascading Style Sheets (CSS) είναι μία style language η οποία χρησιμοποιείται για να περιγράψει την εμφάνιση ενός δομημένου εγγράφου όπως η HTML. Με αυτόν τον τρόπο πετυχαίνετε μία ομοιομορφία στην εμφάνιση και στην διαμόρφωση του layout.

- Το παράδειγμα που ακολουθεί είναι από το css αρχείο της ιστοσελίδας της υπηρεσίας, για την διαμόρφωση των body, form, table και link στοιχείων.

```
body {  
    margin: 0;  
    background-color: #ffffff;  
}  
  
form {  
    font-style : italic;  
}  
  
table {  
    empty-cells : show;
```

```
}  
  
a:link {  
    color : #0887b1;  
    font-family : sans-serif, Arial;  
    font-size : 13px;  
    text-decoration: none;  
}
```

3 ΥΠΑΡΧΟΝΤΑ ΕΡΓΑΛΕΙΑ

Σε αυτό το κεφάλαιο θα αναφερθούμε σε υπάρχοντα εργαλεία που έχουν δημιουργηθεί για την ενημέρωση σχετικά με την επισκεψιμότητα σε έναν web server. Αυτά τα εργαλεία προσφέρουν counters και αναλυτικά στατιστικά μεγέθη για τον αριθμό των επισκέψεων αλλά και το πότε πραγματοποιήθηκαν. Τα παρακάτω εργαλεία δεν πληρούν τις απαιτήσεις του web server του τμήματος για το λόγο του ότι κρατούν στοιχεία μόνο για το σύνολο ενός webserver και όχι για τον κάθε χρήστη ξεχωριστά.

3.1 AWSTATS

Ένα διαδεδομένο και πολύ ισχυρό εργαλείο για την ανάλυση web στατιστικών είναι το Awstats το οποίο ανήκει στην κατηγορία του ελεύθερου λογισμικού και διανέμεται κάτω από την άδεια της GNU. Η πρώτη του έκδοση βγήκε το έτος 2000 και εξελίσσεται μέχρι και σήμερα. Είναι γραμμένο σε γλώσσα προγραμματισμού Pearl, διατίθεται για όλες τις πλατφόρμες και είναι ανοιχτού κώδικα.

Η ηλεκτρονική διεύθυνση του project είναι :

- <http://awstats.sourceforge.net/>

Τα βασικά του χαρακτηριστικά είναι :

- Λειτουργεί με τον Apache και με τις περισσότερες web/ftp/mail υπηρεσίες.
- Αναγνωρίζει πλειάδα log formats και υποστηρίζει και custom μορφή.
- Ανανεώνεται μέσω CLI αλλά και μέσω browser (CGI).
- Αναφέρει τις χώρες προέλευσης των επισκεψεων βασιζόμενο στην IP διεύθυνση και το domain name.
- Αναφέρει και φιλτράρει robots και spiders.
- Παρέχει ημερήσια, μηνιαία και ετήσια στατιστικά.
- Εξάγει στατιστικά σε pdf μορφή.

3.2 ANALOG

Ένα ακόμη εργαλείο το οποίο αναλύει web logs είναι το Analog, η πρώτη του έκδοση ανακοινώθηκε το 1995. Διατίθεται δωρεάν και υπάρχει η δυνατότητα κατεβάσματος των εκτελέσιμων αρχείων, του πηγαίου κώδικα και του το documentation. Η εφαρμογή είναι γραμμένη σε γλώσσα προγραμματισμού C και είναι διαθέσιμη για όλα τα λειτουργικά συστήματα. Επίσης έχει την δυνατότητα να παρουσιάζει αποτελέσματα σε τριανταδύο διαφορετικές γλώσσες.

Η ηλεκτρονική διεύθυνση του project είναι :

- <http://www.analog.cx>

Τα βασικά του χαρακτηριστικά είναι :

- Λειτουργεί με τον Apache web server.
- Αναγνωρίζει πλειάδα log formats και υποστηρίζει και custom μορφή.
- Ανανεώνεται μέσω CLI αλλά και μέσω browser (CGI).
- Αναφέρει τις χώρες προέλευσης των επισκεψεων βασιζόμενο στο domain name.
- Αναφέρει και φιλτράρει robots και spiders.
- Παρέχει ημερήσια, μηνιαία και ετήσια στατιστικά.

3.3 WEBALIZER

Την τριάδα των πιο διαδεδομένων εργαλείων web στατιστικών κλείνει το Webalizer. Είναι και αυτό ελεύθερο λογισμικό ανοιχτού κώδικα και είναι γραμμένο σε γλώσσα προγραμματισμού C. Έχει επιπλέον την δυνατότητα να αναλύει αρχεία εγγραφών web logs, τα οποία είναι συμπιεσμένα σε μορφή αρχείων gzip και bzip2 απευθείας χωρίς να τα αποσυμπιέζει. Τα αποτελέσματα που παράγονται μπορούν να παρουσιαστούν σε περισσότερες από τριάντα διαφορετικές γλώσσες μεταξύ τους και τα ελληνικά.

Η ηλεκτρονική διεύθυνση του project είναι :

- <http://www.mrunix.net/webalizer/>

Τα βασικά του χαρακτηριστικά είναι :

- Λειτουργεί με τον Apache και άλλες web/ftp υπηρεσίες.
- Αναγνωρίζει πλειάδα log formats.

- Αναnevώνεται μέσω CLI.
- Αναφέρει τις χώρες προέλευσης των επίσκεψων βασιζόμενο στο domain name.
- Παρέχει ημερήσια, μηνιαία και ετήσια στατιστικά.

3.4 ΜΕΤΡΗΤΕΣ (COUNTERS)

Παρακάτω παρουσιάζονται υπάρχοντες μετρητές για την τοποθέτηση τους σε ιστοσελίδα με σκοπό την παρουσίαση του αριθμού των επισκέψεων σε έναν ιστότοπο.

1. StatCounter

Η πρώτη εφαρμογή στην οποία θα αναφερθούμε είναι το statcounter. Διατίθεται και μία δωρεάν έκδοση και παρέχει μετρητές επισκέψεων και αναλυτικές πληροφορίες σχετικά με τις επισκέψεις σε αληθινό χρόνο.

Η εγκατάσταση του είναι εύκολη και πολύ απλή. Τα βήματα αυτά είναι η δημιουργία ενός λογαριασμού στην ιστοσελίδα της εφαρμογής και η εισαγωγή κώδικα στην ιστοσελίδα που θέλουμε να παρακολουθήσουμε. Τέλος μένει η ανάλυση των στατιστικών.

2. Addfreestats

Το Addfreestats είναι ένα ελεύθερο εργαλείο το οποίο παρέχει στους webmasters πληροφορίες για τον αριθμό των επισκέψεων αλλά και στοιχεία σχετικά με τους επισκέπτες αλλά και πώς οδηγήθηκαν στην ιστοσελίδα σας. Όλα τα στοιχεία αναλύονται σε πραγματικό χρόνο.

Κάποια από τα στοιχεία τα οποία μας προσφέρει αυτό το εργαλείο είναι ένας μετρητής των συνολικών επισκέψεων αλλά και ένας ακόμα για το πόσοι είναι συνδεδεμένοι την συγκεκριμένη χρονική στιγμή.

Για την εγκατάσταση του το μόνο που χρειάζεται είναι η πρόσθεση κώδικα στην ιστοσελίδα.

3. GoStats

Το Gostats είναι ένα ελεύθερο εργαλείο το οποίο πέραν των μετρητών παρέχει πληροφορίες και για τις εμφανίσεις των ιστοσελίδων ενός ιστότοπου, τους επισκέπτες και την επισκεψιμότητα για μία χρονική περίοδο της επιλογής μας.

Παρέχει σχεδιαγράμματα σχετικά με την κίνηση και εμφανίζει στον παγκόσμιο χάρτη τα σημεία από τα οποία έγιναν οι επισκέψεις. Πέραν αυτών προσφέρει διάφορα στατιστικά στοιχεία όπως μέγιστες και ελάχιστες επισκέψεις και εμφανίζει την κατάταξη των ιστοσελίδων του ιστότοπου, με τις περισσότερες επισκέψεις.

Η εγκατάσταση του είναι ίδια με των προηγούμενων εργαλείων, απλώς προσθέτετε ένα κομμάτι κώδικα στην ιστοσελίδα.

4. Site Meter

Το Site Meter είναι μία υπηρεσία στατιστικών web κίνησης. Η basic έκδοση του διατίθεται δωρεάν και ενημερώνει αναλυτικά σχετικά με τις επισκέψεις, τους επισκέπτες και δίνει την δυνατότητα της εμφάνισης ενός counter σε ιστοσελίδα. Επίσης παρουσιάζει στοιχεία και στατιστικά μεγέθη σχετικά με τις επισκέψεις στον ιστότοπο.

Η εφαρμογή διαχωρίζει τις εμφανίσεις ιστοσελίδων από την επίσκεψη στον ιστότοπο και ενημερώνει για αυτές το ίδιο αναλυτικά με τις επισκέψεις.

4 ΔΟΜΗ / ΑΝΑΛΥΣΗ

Η δομή της εφαρμογής webstats είναι πολύ απλή και αποτελείται από τρία στοιχεία. Σε αυτό το κεφάλαιο παρουσιάζονται και στη συνέχεια αναλύονται το καθένα από αυτά.

- **Parser :** Ο parser είναι το πρόγραμμα που σαρώνει το αρχείο access.log του Apache το οποίο περιέχει εγγραφές με τις επισκέψεις στον web server. Έτσι αντλούνται όλες αυτές οι πληροφορίες που χρειάζονται για τον υπολογισμό στατιστικών για την επισκεψιμότητα στον aetos. Στη συνέχεια όλες αυτές οι εγγραφές αποθηκεύονται σε μία βάση δεδομένων. Οι ενημερώσεις και οι διαγραφές στη βάση εκτελούνται από τον parser. Η εκτέλεση του parser γίνεται από ένα σενάριο φλοιού Unix το οποίο εκτελείται ανά τακτά χρονικά διαστήματα με τη βοήθεια ενός scheduler προγράμματος, του Cron.
- **Web Interface :** Η διασύνδεση της υπηρεσίας με τον χρήστη γίνεται μέσω μιας δυναμικής ιστοσελίδας που δημιουργήθηκε για την παρουσίαση στατιστικών. Αυτά είναι είτε κάποια αριθμητικά μεγέθη είτε σχεδιαγράμματα τα οποία προκύπτουν από ερωτήματα που εκτελούνται στη βάση μετά από σύνδεση του web interface με αυτή.
- **Βάση Δεδομένων :** Η βάση ουσιαστικά είναι το ενδιάμεσο και συνδεδεμένο στοιχείο σε αυτά που αναφέρθηκαν παραπάνω. Στους πίνακες της βάσης είναι αποθηκευμένες όλες οι απαιτούμενες πληροφορίες όπως οι λογαριασμοί των χρηστών του server και οι επισκέψεις για ορισμένα χρονικά διαστήματα. Σε αυτήν συνδέονται ο parser και το web interface, το πρώτο για να την ενημερώσει και το δεύτερο για την εκτέλεση ερωτημάτων για τον υπολογισμό στατιστικών.

4.1 ΓΛΩΣΣΕΣ ΥΛΟΠΟΙΗΣΗΣ

Για την υλοποίηση αυτών των τριών δομικών στοιχείων που αναφέρθηκαν παραπάνω χρησιμοποιήθηκαν εργαλεία όπως γλώσσες προγραμματισμού και συστήματα διαχείρισης βάσεων δεδομένων. Αυτά παρουσιάζονται παρακάτω.

4.1.1 Parser

Η υλοποίηση του Parser έγινε με τη γλώσσα προγραμματισμού python. Η γλώσσα αυτή είναι multiplatform, το συντακτικό της είναι αρκετά απλό και λιτό χαρακτηριστικό που την κάνει αρκετά εύχρηστη και εύκολη στην εκμάθησή της. Είναι μία γλώσσα που μπορεί να υποστηρίξει

πολλά προγραμματιστικά στυλ και είναι πολύ γρήγορη. Οι παραπάνω λόγοι την καθιστούν μία πρακτική γλώσσα και αν και είναι υψηλού επιπέδου ενδείκνυται για πιο low-level εφαρμογές.

Για την σύνδεση της python με την PostgreSQL χρησιμοποιήθηκε η βιβλιοθήκη pyPgSQL η οποία μας προσφέρει ένα interface μεταξύ αυτών των δύο ώστε να μπορούμε να εκτελούμε εισαγωγές, ενημερώσεις και διαγραφές στην βάση από τον parser. Για την χρήση αυτής της βιβλιοθήκης θα πρέπει η έκδοση της PostgreSQL να είναι η 7.0 ή νεότερη ώστε να είναι δυνατή η σύνδεση.

4.1.2 Βάση Δεδομένων

Όσον αφορά την βάση δεδομένων χρησιμοποιήσαμε την PostgreSQL, ένα σύστημα διαχείρισης βάσεων δεδομένων ανοιχτού κώδικα. Αυτή η υπηρεσία είναι διαθέσιμη στο server του τμήματος και στο διαδίκτυο υπάρχει μία μεγάλη κοινότητα που το υποστηρίζει και προσφέρει σημαντικά βοηθήματα.

Αρχικά υπήρξε σαν εναλλακτική επιλογή η Sqlite η οποία τρέχει τοπικά, χρησιμοποιεί ελάχιστους υπολογιστικούς πόρους, είναι πολύ γρήγορη και υπάρχει και η δυνατότητα σύνδεσης της με την python. Το γεγονός όμως ότι δεν υπάρχει η δυνατότητα να λειτουργήσει δικτυακά, απέτρεψε την επιλογή της.

4.1.3 Web Interface

Το layout της ιστοσελίδας διαμορφώθηκε χρησιμοποιώντας stylesheets ώστε να υπάρχει ομοιομορφία στην εμφάνιση.

Για το side server programming χρησιμοποιήθηκε η PHP, μία γλώσσα που διατίθεται δωρεάν και δίνει την δυνατότητα δημιουργίας ιστοσελίδων με δυναμικό περιεχόμενο, επίσης είναι διαθέσιμο ένα πλούσιο documentation στο διαδίκτυο. Για την δημιουργία διασύνδεσης μεταξύ της PHP και της PostgreSQL χρησιμοποιήθηκε η βιβλιοθήκη php-pgsql.

Η εμφάνιση των αποτελεσμάτων που προκύπτουν από την εκτέλεση των php αρχείων έγινε με τη χρήση του Smarty, μία μηχανή templates για php αρχεία. Έτσι διαχωρίστηκε το λειτουργικό μέρος από αυτό της εμφάνισης.

4.2 ΣΧΗΜΑ ΒΑΣΗΣ

Το πρώτο πράγμα που έπρεπε να σχεδιαστεί ουσιαστικά ήταν το σχήμα της βάσης. Πάνω σε αυτό θα προσαρμοζόταν και ο parser εφόσον αυτό το αρχείο θα εκτελούσε τις εισαγωγές, τις ενημερώσεις και της διαγραφές. Τα διακριτά αντικείμενα που υπήρχαν ήταν δύο, οι λογαριασμοί του aetos που είχαν ανεβάσει ιστοσελίδες και οι επισκέψεις. Για καλύτερη χρήση της πληροφορίας αλλά και για να μειωθεί ο όγκος της βάσης ο πίνακας επισκέψεις διασπάστηκε σε τρεις.

Αυτοί οι πίνακες είναι ο επισκέψεις ανά ώρα, ανά μέρα και επισκέψεις ανά μήνα. Οι πρώτοι δύο κρατάνε εγγραφές για ένα μικρό χρονικό διάστημα, τόσο ώστε να μπορεί να παρουσιαστεί

η κίνηση τις τελευταίες ώρες και μέρες αντίστοιχα. Η διάσπαση στον πίνακα των επισκέψεων έγινε αναλύοντας το πεδίο της ημερομηνίας, έτσι επετεύχθη μία καλύτερη χρήση της πληροφορίας αποφεύγοντας χρονοβόρες προσπελάσεις και πράξεις για το σύστημα διαχείρισης της βάσης και μειώθηκε σημαντικά ο όγκος. Επιπλέον δημιουργήθηκε ένας ακόμα πίνακας που κρατάει εγγραφές για τις επισκέψεις ανά ώρα και μήνα για κάθε λογαριασμό ώστε να υπάρχει δυνατότητα να υπολογίζεται η κατανομή των επισκέψεων ανά ώρα σε βάθος εώς και ενός χρόνου.

Ο πίνακας `usrdir` κρατάει εγγραφές με τους λογαριασμούς που έχουν ανεβάσει ιστοσελίδες στο server και έχει τα παρακάτω πεδία.

- Πρωτεύον κλειδί είναι το `id`, είναι ένα τεχνητό κλειδί που παίρνει μοναδικές τιμές από ένα `sequence` που δημιουργήθηκε στη βάση και είναι ο αριθμός ταυτότητα του λογαριασμού.
- Το πεδίο `uname` που αντιστοιχεί στο όνομα του λογαριασμού.
- Το πεδίο `directory` για το `path` του `home directory` του λογαριασμού.
- Το πεδίο `group` για την κατηγορία που ανήκει ο λογαριασμός.

Αν και κάθε φορά υπάρχει μόνο ένας λογαριασμός που αντιστοιχεί σε κάποιο όνομα επειδή υπάρχει η περίπτωση όπου ένας λογαριασμός διαγράφεται και δημιουργείται ένας καινούργιος με το ίδιο όνομα για αυτό το λόγο δεν δηλώνεται `UNIQUE`. Βέβαια όταν ένας λογαριασμός έχει διαγραφεί στην επόμενη εκτέλεση του `parser` θα διαγραφεί και από τη βάση όμως ίσως υπάρξουν κάποιες ακραίες περιπτώσεις όπου θα συνυπάρξουν αυτοί οι δύο λογαριασμοί στη βάση.

Όσον αφορά το πεδίο `directory` δηλώνεται `UNIQUE` για ασφάλεια αφού είναι μοναδικό για κάθε λογαριασμό λόγω της οργάνωσης των λογαριασμών στο server του τμήματος. Με αυτήν τη δήλωση είναι σαν να έχουμε ένα ακόμα πεδίο κλειδί για τις εγγραφές του πίνακα `usrdir`.

Τέλος έχουμε το πεδίο `ugroup` που αντιπροσωπεύει την ομάδα του λογαριασμού. Οι λογαριασμοί στον `aetos` χωρίζονται σε `staff` και `students`, δηλαδή σε αυτές του προσωπικού του τμήματος όπως οι καθηγητές και σε αυτές των φοιτητών. Προσθέσαμε αυτό το πεδίο για να έχουμε την δυνατότητα να υπολογίσουμε κάποια στατιστικά μεγέθη σχετικά με αυτές τις κατηγορίες. Πέραν αυτών των δύο δημιουργήθηκε μία επιπλέον, η κατηγορία `other` για να εντάξουμε σε αυτήν το `root document` του server, δηλαδή τη σελίδα του τμήματος αλλά και για να καλύψουμε κάποια νέα κατηγορία που ίσως προκύψει στο μέλλον. Σε ποιά κατηγορία ανήκει ένας λογαριασμός προκύπτει από το `home directory` τους βάση της πολιτικής που εφαρμόζεται για την οργάνωση και κατηγοριοποίηση των λογαριασμών στο server του τμήματος.

Οι πίνακες `hpm`, `hpd`, `hph`, δηλαδή αυτοί που κρατούν εγγραφές με τις επισκέψεις ανά μήνα, μέρα και ώρα αντίστοιχα έχουν τα πεδία `hits`, δηλαδή των αριθμό των επισκέψεων και `data` για τον αριθμό των δεδομένων που έχουν κατέβει από τις επισκέψεις. Τον όγκο των δεδομένων των κρατάμε σε `Kilobytes`. Για τον πίνακα που κρατάει εγγραφές ανά μήνα υπάρχουν επιπλέον τα πεδία `month` για το μήνα και `year` για τον χρόνο. Για τον ανά ώρα και μέρα έχουμε το

πεδίο date για την ημερομηνία. Επιπλέον υπάρχει το πεδίο hour για την ώρα στο πίνακα με τις επισκέψεις ανά ώρα.

Αυτοί οι τρεις πίνακες έχουν ακόμα τα πεδία uid και ts. Το uid είναι ξένο κλειδί και αναφέρεται στο πεδίο id του πίνακα usrdir. Χρησιμοποιείτε για την δημιουργία συσχέτισης ένα προς πολλά μεταξύ του πίνακα των χρηστών και των πινάκων των επισκέψεων, έτσι ώστε να αντιστοιχηθεί μία επίσκεψη με τον ιστότοπο ενός λογαριασμού. Το πεδίο ts είναι τύπου timestamp και είναι το κριτήριο για την διαγραφή των εγγραφών, εφόσον η διάρκεια ζωής τους έχει λήξει. Οι εγγραφές του πίνακα των επισκέψεων ανά μήνα έχουν διάρκεια πέντε χρόνων αυτή η χρονική διάρκεια θεωρήθηκε αρκετή για την εξαγωγή ασφαλών συμπερασμάτων, αυτό είναι και το μέγιστο βάθος χρόνου στατιστικών στοιχείων της υπηρεσίας. Για τους άλλους δύο πίνακες, ανά μέρες και ώρες, οι εγγραφές έχουν διάρκεια τριανταμία μέρες και εικοσιτέσσερις ώρες αντίστοιχα. Αυτοί χρησιμοποιούνται για την παρουσίαση της κίνησης το τελευταίο χρονικό διάστημα γι' αυτό και η τόσο μικρή διάρκεια ζωής.

Πέραν αυτών των τριών πινάκων για τις επισκέψεις υπάρχει ακόμα ένας με το όνομα hhm. Η χρήση του είναι για τον υπολογισμό της κατανομής των επισκέψεων ανά ώρα. Τα πεδία που έχει είναι το uid (που αναφέρεται στον πίνακα usrdir), πεδία για τον μήνα, έτος, ώρα, αριθμό επισκέψεων και ένα πεδίο τύπου timestamp για την διαγραφή των εγγραφών παλαιότερων από δώδεκα μήνες. Αυτός ο πίνακας δίνει την δυνατότητα της ενημέρωσης για τον τρόπο που κατανέμεται η επισκεψιμότητα στην διάρκεια της ημέρας σε βάθος έως και ενός χρόνου.

Τέλος δημιουργήθηκε μια αυξανόμενη κατά ένα ακολουθία (sequence) με το όνομα seq_usrdir_id για να δίνει μοναδικές τιμές στο πεδίο id, κλειδί του πίνακα usrdir.

4.2.1 Πίνακες Βάσης

Παρακάτω παρουσιάζονται οι πίνακες της βάσης.

USRDIR (λογαριασμοί χρηστών του server)

| id | integer | πρωτεύων κλειδί |
|-----------|------------|-----------------|
| uname | varchar | |
| directory | varchar | unique |
| ugroup | varchar(7) | |

HPM (επισκέψεις ανά μήνα)

| | | | |
|-------|-----------|--------|---------------------------|
| uid | integer | κλειδί | αναφέρεται στο usrdir(id) |
| month | integer | κλειδί | |
| year | integer | κλειδί | |
| hits | integer | | |
| data | integer | | |
| ts | timestamp | | |

HPD(επισκέψεις ανά μέρα)

| | | | |
|------|-----------|--------|---------------------------|
| uid | integer | κλειδί | αναφέρεται στο usrdir(id) |
| date | date | κλειδί | |
| hits | integer | | |
| data | integer | | |
| ts | timestamp | | |

HPH (επισκέψεις ανά ώρα)

| | | | |
|------|-----------|--------|---------------------------|
| uid | integer | κλειδί | αναφέρεται στο usrdir(id) |
| hour | integer | κλειδί | |
| date | date | κλειδί | |
| hits | integer | | |
| data | integer | | |
| ts | timestamp | | |

HHM (κατανομή των επισκέψεων ανά ώρα)

| | | | |
|-------|-----------|--------|---------------------------|
| uid | integer | κλειδί | αναφέρεται στο usrdir(id) |
| hour | integer | κλειδί | |
| month | integer | κλειδί | |
| year | integer | κλειδί | |
| hits | integer | | |
| ts | timestamp | | |

Τα πεδία που συμμετέχουν στα σύνθετα κλειδιά δηλώθηκαν NOT NULL. Επίσης για τα πεδία hour και month δημιουργήθηκαν constraints ώστε να αποτρέπεται μία μη αποδεκτή τιμή. Επίσης δημιουργήθηκε constraint για το πεδίο ugroup ώστε να δέχεται τις αποδεκτές τιμές staff, student και other.

Για τα ξένα κλειδιά που αναφέρονται στο id του πίνακα usrdir δηλώθηκαν cascades έτσι ώστε να επηρεάζονται από τις μεταβολές που θα γίνονται στην εγγραφή που αναφέρονται.

4.2.2 SQL κώδικας δημιουργίας των tables

Παρακάτω παρατίθεται ο SQL κώδικας που δημιουργεί τους πίνακες στη βάση. Ο κώδικας αυτός είναι το περιεχόμενο του αρχείου webstats_tables.sql.

```
1 CREATE TABLE usrdir (  
2     id INTEGER PRIMARY KEY,  
3     uname VARCHAR,  
4     directory VARCHAR UNIQUE,  
5     ugroup VARCHAR(7) CONSTRAINT c_ugroup  
6         CHECK (ugroup in ('staff', 'student', 'other'))  
7 );  
  
1 CREATE TABLE hph (  
2     uid INTEGER REFERENCES usrdir(id)  
3         ON UPDATE CASCADE  
4         ON DELETE CASCADE,  
5     hour INTEGER NOT NULL CONSTRAINT c_hour  
6         CHECK (hour >= 0 AND hour < 24),  
7     date DATE NOT NULL,  
8     hits INTEGER,  
9     data INTEGER,  
10    ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
11    PRIMARY KEY (uid, hour, date)  
12 );
```

```
1 CREATE TABLE hpd (  
2     uid INTEGER REFERENCES usrdir(id)  
3         ON UPDATE CASCADE  
4         ON DELETE CASCADE,  
5     date DATE NOT NULL,  
6     hits INTEGER,  
7     data INTEGER,  
8     ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
9     PRIMARY KEY (uid, date)  
10 );
```

```
1 CREATE TABLE hpm (  
2     uid INTEGER REFERENCES usrdir(id)  
3         ON UPDATE CASCADE  
4         ON DELETE CASCADE,  
5     month INTEGER NOT NULL CONSTRAINT c_month  
6         CHECK (month>0 AND month<=12),  
7     year INTEGER NOT NULL,  
8     hits INTEGER,  
9     data INTEGER,  
10    ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
11    PRIMARY KEY (uid, month, year)  
12 );
```

```
1 CREATE TABLE hhm (  
2     uid INTEGER REFERENCES usrdir(id)  
3         ON UPDATE CASCADE  
4         ON DELETE CASCADE,  
5     month INTEGER NOT NULL CONSTRAINT c2_month  
6         CHECK (month>0 AND month<=12),  
7     hour INTEGER NOT NULL CONSTRAINT c2_hour  
8         CHECK (hour>=0 AND hour<24),  
9     year INTEGER NOT NULL,  
10    hits INTEGER,  
11    ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
12    PRIMARY KEY (uid, month, year, hour)  
13 );
```

Τέλος ο κώδικας για τη δημιουργία μιας ακολουθίας για την απόδοση τιμής στο id, το πεδίο κλειδί του πίνακα usrdir.

```
CREATE SEQUENCE seq_usrdir_id;
```

4.2.3 Σχεδιάγραμμα ER της Βάσης

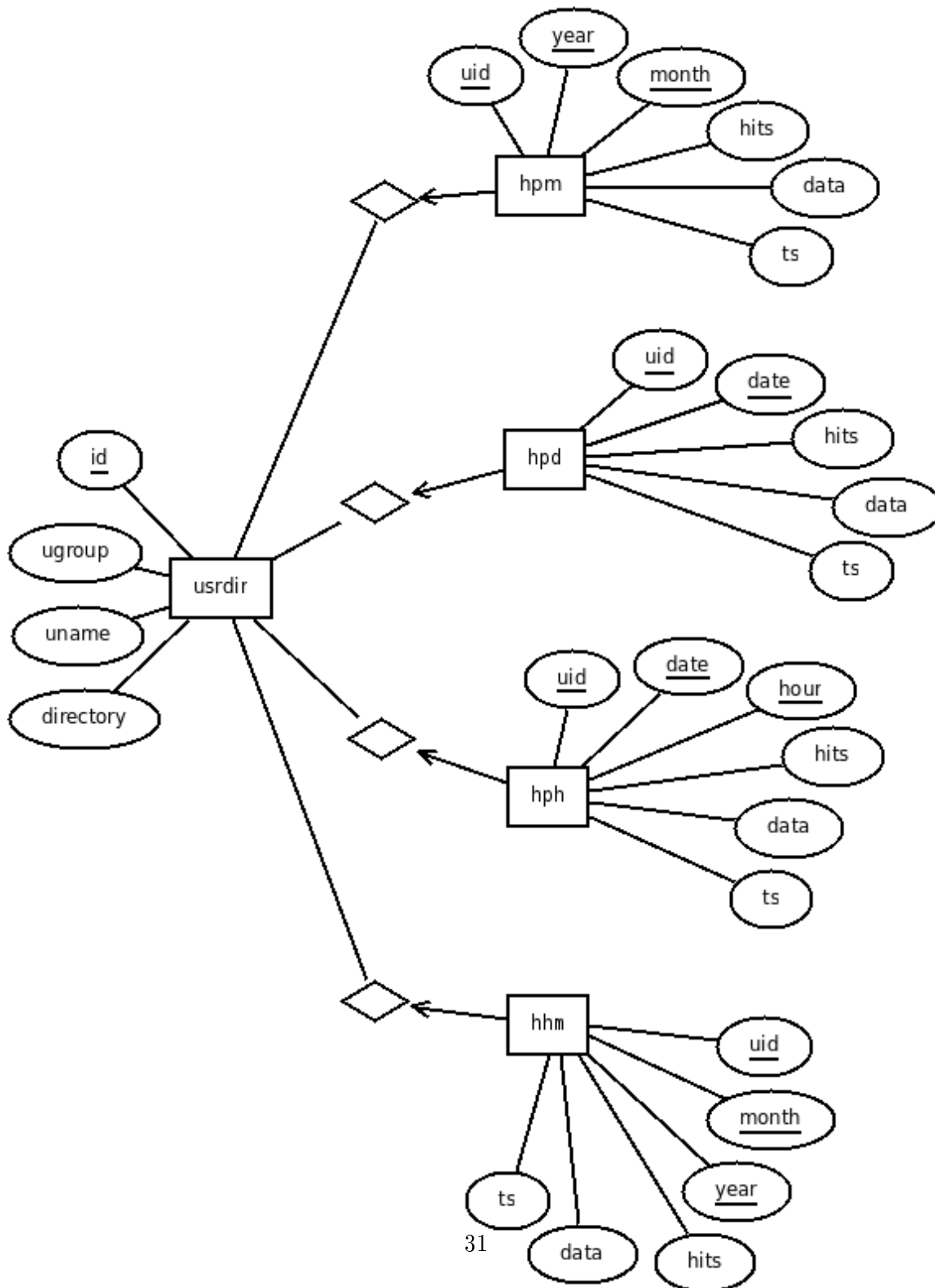


Figure 4.1: Σχεδιάγραμμα ER

5 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ

5.1 LAUNCHER.SH

Το αρχείο `launcher.sh` είναι ένα σενάριο φλοιού το οποίο εκτελείτε προγραμματισμένα κάθε δεκαπέντε λεπτά με την χρήση του `Cron`. Έτσι παραμένει ενήμερη η βάση δεδομένων της υπηρεσίας για τις νέες επισκέψεις που πραγματοποιήθηκαν.

Αρχικά εκτελείτε η `logtail` η οποία διαβάζει τις νέες εγγραφές του αρχείου `access.log` που περιέχει τα `logs` των επισκέψεων, το αποτέλεσμα αυτής της ενέργειας διοχετεύεται σε ένα αρχείο με το όνομα `feed`. Στην συνέχεια εκτελείται από το σενάριο ο `parser` με παράμετρο το αρχείο `feed`. Με αυτόν τον τρόπο γίνεται η επεξεργασία των νέων εγγραφών του `access log` αρχείου του `apache` και παραμένει ενήμερη η βάση.

5.1.1 Κώδικας του `launcher.sh`

```
1 #!/bin/bash
2 echo "for aetos"
3 echo "...fishing data"
4 logtail -faetos-access.log -oaetos_offset > feed
5 parser.py feed
6 echo "for www"
7 echo "...fishing data"
8 logtail -fwww-access.log -owww_offset > feed
9 parser.py feed
10 echo "JOB IS DONE!"
```

Παρατηρούμε ότι γίνεται η ίδια διαδικασία δύο φορές. Μία φορά για το αρχείο `aetos-access.log` και μία φορά για το αρχείο `www-access.log`. Ο λόγος είναι ότι στην περίπτωση του `aetos` ο `web server` ακούει σε δύο `domain names` και έτσι ο `apache` κρατάει στοιχεία σε δύο διαφορετικά αρχεία, ένα για κάθε όνομα.

5.2 PARSER.PY

Ο `parser` είναι ένα `python script` όπου παίρνει ως παράμετρο ένα αρχείο με `apache access logs` σε `common` ή σε `combined` μορφή. Η λειτουργία του δεν είναι άλλη από το να διαβάζει τα `logs`,

να ενημερώνει την βάση και στη συνέχεια να καθαρίζει τους πίνακες από τις παλιές εγγραφές.

Αρχικά θα γίνει αναφορά στις συναρτήσεις του parser, τις παραμέτρους τους και το τι επιστρέφουν.

- `month_to_num` : Διαβάζει ένα string το οποίο είναι οι τρεις πρώτοι χαρακτήρες ενός μήνα, με την μορφή που εμφανίζονται στα logs και επιστρέφει τον αριθμό που τον αντιπροσωπεύει. πχ `month_to_num('Dec')` returns 12.

```
1 def month_to_num(x):
2     months = {
3         'Jan' : 1,
4         'Feb' : 2,
5         'Mar' : 3,
6         'Apr' : 4,
7         'May' : 5,
8         'Jun' : 6,
9         'Jul' : 7,
10        'Aug' : 8,
11        'Sep' : 9,
12        'Oct' : 10,
13        'Nov' : 11,
14        'Dec' : 12
15    }
16    return months[x];
```

- `parseDates` : Αυτή η συνάρτηση παίρνει σαν παράμετρο ένα access log του apache και επιστρέφει την ημερομηνία και την ώρα τις επίσκεψης αναλυτικά.

```
1 def parseDates(x):
2     date = x[ (x.find('[')+1) : (x.find('')+12) ];
3     year = int(x[ (x.find('')+8) : (x.find(':')) ]);
4     month = month_to_num(x[ (x.find('[')+4) : (x.find('')+7) ]);
5     hour = x[ (x.find(':')+1) : (x.find('')+9) ];
6     date_hour = date+'_'+hour;
7     hour = int(hour[0:2]);
8     ret = (date, year, month, hour, date_hour);
9     return ret;
```

- `getUname` : Οι παράμετροι αυτής της συνάρτησης είναι ένα `log` και ένας `integer` που αντιπροσωπεύει την θέση του `http code` στο `log`. Αυτός ο αριθμός χρειάζεται για να μην παίρνεται υπόψη ο `home` χαρακτήρας που ενδεχόμενος να προέρχεται από το πεδίο του `referer`. Η συνάρτηση επιστρέφει το όνομα του λογαριασμού του `server` στο οποίο την ιστοσελίδα έγινε η επίσκεψη.

```

1 def getUname(x, http_code_pos):
2     home_char_pos = x.find('~');
3     if home_char_pos == -1 or home_char_pos > http_code_pos:
4         //apodosi timis gia tin kentriki selida
5         usr = ':main';
6     else:
7         usr = x[ (x.find('~') + 1) : ];
8         usr = usr[ 0 : usr.find('/') ];
9         if usr.find('_') != -1:
10            usr=usr[0 : usr.find('_')];
11    return usr;
```

- `getDir` : Αυτή η συνάρτηση παίρνει σαν παράμετρο ένα `username` και επιστρέφει το `home directory` του.

```

1 def getDir(usr):
2     if usr == ':main':
3         directory = '/var/www';
4     else:
5         directory = pwd.getpwnam(usr).pw_dir;
6     return directory;
```

- `getGroup` : Εδώ η παράμετρος είναι το `home directory` ενός `account` και επιστρέφει το `group` που ανήκει ο χρήστης.

```

1 def getGroup(directory):
2     if directory.find('staff') != -1:
3         group = 'staff';
4     elif directory.find('student') != -1:
5         group = 'student';
6     else:
7         group = 'other';
8     return group;
```

- `getData` : Η παράμετρος είναι ένα log και επιστρέφει τον αριθμό των δεδομένων τα οποία μεταφέρθηκαν από αυτήν την αίτηση της επίσκεψης.

```

1 def getData(x):
2     r = re.compile('[2-5][0-9][0-9]');
3     http_code = r.search(x).group();
4     http_code_pos = x.find(http_code);
5     if http_code == '200':
6         data = x[x.find(http_code)+5 : ];
7         try:
8             data = int(data[0 : data.find(',')]);
9         except:
10            data = 0;
11    else:
12        data = 0;
13    if data != 0:
14        data = data / 1024.0;
15    http_code = int(http_code.strip());
16    ret = (data, http_code, http_code_pos);
17    return ret;

```

- `checkSpam` : Η συγκεκριμένη συνάρτηση παίρνει σαν παράμετρο ένα access log και ένα πίνακα που περιέχει regular expressions για spiders και robots μηχανών αναζήτησης. Στη συνέχεια ελέγχεται το log εάν προέρχεται από κάποιο robot και αν ναι απορρίπτεται.

```

1 def checkSpam(x, black_list):
2     check = False;
3     for i in black_list:
4         r=re.compile(i);
5         try:
6             tmp = r.search(x).group();
7         except:
8             tmp = -1;
9         if tmp != -1:
10            check = True;
11    return check;

```

Αφού ξεκινήσει η εκτέλεση του parser, φορτώνονται τα πακέτα `sys`, `re` και `pwd` από την standard βιβλιοθήκη της python. Το πακέτο `sys` το χρησιμοποιούμε για να μπορεί το πρόγραμμα να παίρνει σαν παράμετρο το αρχείο που πρόκειται να επεξεργαστεί από την κονσόλα. Με το

πακέτο `re` έχουμε την δυνατότητα να κάνουμε `match` τα `logs` με κάποια `patterns` που έχουν δημιουργηθεί με τη χρήση των `regular expressions`. Το πακέτο `pwd` μας παρέχει πληροφορίες για ένα λογαριασμό, όπως το `login name`, έτσι δίνεται η δυνατότητα ανάκτησης του `home directory` και άλλων χρήσιμων πληροφοριών για ένα λογαριασμό.

Εκτός των παραπάνω φορτώνεται και το πακέτο `pyPgsql` που δεν ανήκει στην `standard` βιβλιοθήκη της `python` και θα πρέπει να έχει εγκατασταθεί στο σύστημα. Αυτό το πακέτο δημιουργεί μια γέφυρα επικοινωνίας μεταξύ της `python` και της `PostgreSQL`, έτσι επιτυγχάνεται η ενημέρωση της βάσης μέσα από τον `python` κώδικα.

Αφού γίνουν τα `imports` ο `parser` εκτελεί κάποιες περαιτέρω λειτουργίες, αυτές είναι η σύνδεση στην βάση και η φόρτωση του αρχείου που περιέχει τα `logs`. Αυτό το αρχείο δίνεται ως παράμετρος από την `κονσόλα` όταν εκτελείται ο `parser`. Εν συνεχεία φορτώνονται στην μνήμη τα στοιχεία των λογαριασμών από τον πίνακα `usrdir` της βάσης.

- Ο κώδικας για την σύνδεση στην βάση :

```
1 con = PgSQL.connect (
2     database=dbname ,
3     user=dbuser ,
4     password=dbpasswd
5 );
```

- Ο κώδικας για να διαβαστούν οι λογαριασμοί των χρηστών από την βάση :

```
1 cur = con.cursor ();
2 cur.execute("SELECT id, uname, directory, ugroup
3             FROM usrdir;")
4 );
5 res = cur.fetchone ();
```

Έχοντας εκτελέσει όλες τις απαραίτητες προεργασίες ο `parser` ξεκινάει μία επανάληψη εώς ότου διαβάσει όλα τα `logs` του αρχείου. Αυτή η επεξεργασία γίνεται εμφωλευμένη σε ένα `try` για θέματα ασφάλειας. Αν κάποιο `log` δεν ταιριάζει σε κάποιο `pattern` να μην σταματήσει η λειτουργία του προγράμματος και να συνεχιστεί εώς ότου σαρωθούν όλες οι εγγραφές του αρχείου. Σε αυτήν τη επεξεργασία εξετάζονται τα `logs` και χρησιμοποιώντας τις συναρτήσεις που αναφέρθηκαν παραπάνω αποκτούνται όλες οι πληροφορίες που θα χρειάζονται για την δημιουργία `dictionaries`, μίας δομής δεδομένων της `python`, που αντιστοιχούν σε εγγραφές των πινάκων της βάσης.

Για την εγγραφή στη βάση μιας επίσκεψης, αυτή θα πρέπει να έχει ολοκληρωθεί επιτυχώς, αυτό εξετάζεται από τον `http` κωδικό που υπάρχει σε κάθε `log`. Αν η επίσκεψη είναι επιτυχημένη τότε ο κωδικός θα ισούται με `200`. Πέραν της εξέτασης του `http` κωδικού εξετάζεται και αν η επίσκεψη προέρχεται από κάποιο `robot` ή `spider`. Στη συνέχεια τα `dictionaries` προσάπτονται

σε λίστες που αυτές αντιστοιχούν σε πίνακες της βάσης. Δηλαδή υπάρχει μία λίστα για κάθε πίνακα που περιέχει dictionaries αντί για εγγραφές. Ουσιαστικά κρατάμε αυτά τα νέα στοιχεία στην μνήμη για να αποφύγουμε άσκοπες αναφορές στη βάση που έχουν μεγάλο I/O κόστος. Μέτα την ολοκλήρωση του parsing πραγματοποιείται η ενημέρωση της βάσης, έτσι γίνονται μόνο οι απαραίτητες εγγραφές και ενημερώσεις σε αυτήν.

Οι λίστες της rython έχουν ελάχιστα κοινά με την γνωστή δομή δεδομένων. Ουσιαστικά είναι πίνακες μεταβλητού μεγέθους. Η δομή dictionary της rython από την άλλη είναι ένας πίνακας που οι δείκτες του αντί για αριθμοί είναι strings.

Πρέπει να αναφερθεί πως όταν γίνεται μία επίσκεψη σε μία ιστοσελίδα παρατηρούνται πολλά διαδεχόμενα logs από τον ίδιο επισκέπτη, αυτό συμβαίνει γιατί γίνονται εγγραφές και για τα αρχεία που χρησιμοποιούνται από μια ιστοσελίδα όπως φωτογραφίες ή ένα css αρχείο. Για την αποφυγή του να υπολογίζονται σαν επισκέψεις όλα όσα κατεβαίνουν για να εμφανιστεί μια ιστοσελίδα σε ένα browser, θεωρήθηκε το χρονικό διάστημα των τεσσάρων δευτερολέπτων για το οποίο τα διαδεχόμενα logs από τον ίδιο επισκέπτη προς τον ίδιο ιστότοπο δεν υπολογίζονται σαν επισκέψεις και υπόψη λαμβάνεται μόνο ο όγκος των δεδομένων.

Αφου ολοκληρωθεί η ενημέρωση της βάσης απομένει μια τελευταία λειτουργία, αυτή της διαγραφής εγγραφών που έχουν κλείσει τον κύκλο ζωής τους. Αρχικά διαγράφονται οι εγγραφές του πίνακα usridr, δηλαδή όσοι λογαριασμοί έχουν διαγραφεί από το σύστημα ή δεν έχουν καμία αναφορά σε πίνακες με εγγραφές επισκέψεων. Έπειτα διαγράφονται οι εγγραφές από τους πίνακες των επισκέψεων οι οποίες έχουν ξεπεράσει το χρόνο ζωής τους όπως αυτοί αναφέρθηκαν σε προηγούμενη παράγραφο.

- Ο κώδικας για την εισαγωγή εγγραφής στον πίνακα hpm (επισκέψεις ανά μήνα).

```

1 cur.execute("INSERT INTO hpm (uid, year, month, hits, data, ts)
2     VALUES ('%i', '%i', '%s', '%i', '%f', '%s') "
3     % (x['usrid'], x['year'], x['month'],
4     x['hits'], x['data'], x['ts']))
5 );
6 con.commit();
```

- Ο κώδικας για την ενημέρωση εγγραφής στον πίνακα hpm (επισκέψεις ανά μήνα).

```

1 cur.execute("UPDATE hpm
2     SET hits=hits+'%i', data=data+'%f'
3     WHERE uid='%i'
4     AND year='%i'
5     AND month='%s' "
6     % (x['hits'], x['data'], x['usrid'], x['year'], x['month']))
7 );
8 con.commit();
```

- Ο κώδικας για την διαγραφή εγγραφών από την βάση.

```

1 #CLEARING TABLES
2 cur.execute("DELETE FROM usrdir
3             Where id IN(
4                 SELECT u.id
5                 FROM usrdir u
6                 WHERE u.id NOT IN(
7                     SELECT u2.id
8                     FROM usrdir u2, hpm h
9                     WHERE u2.id = h.uid
10                )
11            )"
12 );
13 con.commit();
14 cur.execute("DELETE FROM hph
15             WHERE ts < CURRENT_TIMESTAMP - interval '24 hour'"
16 );
17 cur.execute("DELETE FROM hpd
18             WHERE ts < CURRENT_TIMESTAMP - interval '31 day'"
19 );
20 cur.execute("DELETE FROM hpm
21             WHERE ts < CURRENT_TIMESTAMP - interval '60 month'"
22 );
23 cur.execute("DELETE FROM hhm
24 where ts < CURRENT_TIMESTAMP - interval '12 month'"
25 );
26 con.commit();
27 #END OF CLEAR

```

5.3 WEB INTERFACE

Για τη διασύνδεση του χρήστη με την υπηρεσία δημιουργήθηκε μία ιστοσελίδα στο server. Το layout διαμορφώθηκε χρησιμοποιώντας style sheets. Για ευκολότερη περιήγηση πέραν του βασικού μενού δημιουργήθηκε ένα υπομενού το οποίο εμφανίζει περισσότερες επιλογές σχετικές με αυτό που έχει ζητηθεί. Το υπομενού και το περιεχόμενο της σελίδας εμφανίζονται σε δύο html divs τα οποία δεν έχουν σταθερές διαστάσεις αλλά μεταβάλλονται ανάλογα με το μέγεθος της πληροφορίας που εμφανίζουν, αυτό επετεύχθη με τη χρήση των CSS όπως αναφέρθηκε παραπάνω. Επιπλέον υπάρχει ένα div στο οποίο εμφανίζεται σε ποια ιστοσελίδα βρισκόμαστε.

5.3.1 Προγραμματιστικό μέρος web interface

Όσον αφορά το προγραμματιστικό μέρος της ιστοσελίδας, οι κύριες απαιτήσεις ήταν η σύνδεση με την βάση, η εκτέλεση των κατάλληλων sql queries και τέλος η εμφάνιση των αποτελεσμάτων. Τα δύο πρώτα υλοποιήθηκαν με την χρήση της PHP, για την σύνδεση αυτής με την PostgreSQL χρησιμοποιήθηκε το interface php-postgres. Για την εμφάνιση των αποτελεσμάτων χρησιμοποιήθηκε το Smarty, μια μηχανή που παράγει templates για PHP. Με αυτόν τον τρόπο διαχωρίστηκε το λειτουργικό μέρος από αυτό της εμφάνισης και έτσι ο PHP κώδικας δεν μπλέχτηκε με HTML tags. Όταν καλείται ένα php, αφού εκτελεστεί διοχετεύει τα αποτελέσματα στο Smarty, κατόπιν εμφανίζονται διαμορφωμένα από τα templates που έχουν ορισθεί.

Αναλυτικότερα, καλώντας ένα php αρχείο, αρχικοποιούνται οι μεταβλητές του από τα links χρησιμοποιώντας την http μέθοδο GET. Έπειτα εκτελείται το αρχείο το οποίο συνδέεται στην βάση, στη συνέχεια εξετάζονται οι τιμές των παραμέτρων και διαμορφώνεται το κατάλληλο query το οποίο εκτελείται από το DBMS. Κατόπιν η βάση επιστρέφει τα αποτελέσματα τα οποία διοχετεύονται στο Smarty για να εμφανιστούν διαμορφωμένα από τα templates που έχουν ορισθεί.

- Η εντολή για την σύδεση στην βάση :

```
1 $dbcon = pg_connect ("dbname=$dbname
2                      user=$dbusr
3                      password=$passwd")
4 or die('Could not connect: ' . pg_last_error());
```

- Η εντολή για την εκτέλεση ενός query :

```
1 $result=pg_query($query)
2 or die('Query failed: ' . pg_last_error());
3 $arr1=pg_fetch_all($result);
4 pg_free_result($result); //Free resultset
```

Το λειτουργικό μέρος αποτελείται από πέντε php αρχεία, τα οποία αναφέρονται παρακάτω.

- stats.php : Εμφανίζει στοιχεία σχετικά με την κίνηση της ημέρας όπως των αριθμό των επισκέψεων, των όγκο δεδομένων που κατέβηκαν και μεγέθη σχετικά με την κατανομή των επισκέψεων ανά ομάδα ιστότοπων.
- showdata.php : Αναζητά ένα λογαριασμό στη βάση, εμφανίζει τις συνολικές επισκέψεις και τον συνολικό όγκο που κατέβηκε από ένα λογαριασμού ή μια ομάδα ή για όλο τον server. Επίσης εμφανίζει στοιχεία για την κίνηση σε έναν ιστότοπο για τα χρονικά διαστήματα της τρέχουσας ώρας, μέρα, μήνα, τους τελευταίους τρεις και έξι μήνες, το τρέχων έτος και τα τελευταία δύο έτη.

- `mostvisited.php` : Εμφανίζει τους ιστότοπους με την μεγαλύτερη επισκεψιμότητα στο σύνολο του server αλλά και για κάθε κατηγορία ξεχωριστά. Υπάρχει η δυνατότητα να αναζητήσετε τη θέση ενός λογαριασμού σε αυτή τη λίστα ή ακόμα και να κινηθείτε σε αυτήν με την χρήση βέλων.
- `topmonths.php` : Εμφανίζει τους μήνες με τις περισσότερες επισκέψεις για κάποιον ιστότοπο και για κάθε κατηγορίας.
- `graphs.php` : Δημιουργεί γραφήματα για την παρουσίαση τις επισκεψιμότητας ενός ιστότοπου για συγκεκριμένα χρονικά διαστήματα και υπολογίζει τον μέσο όρο των επισκέψεων για αυτά. Επίσης για την κατανομή των επισκέψεων ανα κατηγορία αλλά και για την παρουσίαση της κατανομής των επισκέψεων στην διάρκεια της μέρας.

Για κάθε ένα από τα παραπάνω αρχεία υπάρχει και το αντίστοιχο template για την μορφοποίηση της εμφάνισης των αποτελεσμάτων τους.

Στο αρχείο `graphs.php` χρησιμοποιείται η βιβλιοθήκη Graphpate για την δημιουργία των γραφημάτων. Με την βοήθεια αυτής της βιβλιοθήκης δημιουργούνται γραφήματα τύπου γραμμής για την εμφάνιση της κατανομής των επισκέψεων ανά ώρα. Όσον αφορά την αναπαράσταση τις κίνησης των τελευταίων έως και εικοσιτεσσάρων ωρών, τριανταενός ημερών, των τελευταίων μηνών και χρόνων παρουσιάζονται με γραφήματα τύπου στήλης. Για την εμφάνιση της κατανομής των επισκέψεων στον webserver ανά κατηγορία χρησιμοποιούνται γράφηματα τύπου πίτας και για αναλυτική παρουσίαση αυτής της κατανομής για συγκεκριμένα χρονικά διαστήματα χρησιμοποιούνται γραφήματα τύπου σωρευμένων στηλών.

- Τα χρονικά αυτά διαστήματα είναι :
 1. Τον τρέχων μήνα
 2. Τους τελευταίους τρεις μήνες
 3. Τους τελευταίους έξι μήνες
 4. Το τρέχων έτος
 5. Τα τελευταία δύο έτη
 6. Συνολικά

Η βιβλιοθήκη Graphpate είναι γραμμένη σε PHP και είναι ανοιχτού κώδικα. Η εξέλιξη της βιβλιοθήκης γίνεται πλέον από την PEAR με το όνομα `Image_Graph`. Η μόνη απαίτηση για τη λειτουργία της είναι η βιβλιοθήκη GD, για καλύτερα αποτελέσματα συνιστάται η δεύτερη έκδοσης της, GD 2. Για να μην επιβαρυνθεί το σύστημα, τα αρχεία της βιβλιοθήκης προστέθηκαν στον κατάλογο της ιστοσελίδας της υπηρεσίας και φορτώνονται μόνο κατά την εκτέλεση του `graphs.php`.

Το αρχείο `Graph` πέραν της δημιουργίας γραφημάτων για την αναλυτική εμφάνιση των επισκέψεις για ένα χρονικό διάστημα, υπολογίζει και εμφανίζει και τον μέσο όρο επισκέψεων για αυτό το διάστημα και τον όγκο δεδομένων που κατέβηκαν από την ιστοσελίδα.

Το web interface της υπηρεσίας υπολογίζει και παρουσιάζει τους ιστότοπους με την μεγαλύτερη επισκεψιμότητα στο server και σε κάθε κατηγορία ξεχωριστά. Αρχικά εμφανίζει τους πρώτους δέκα όμως υπάρχει η δυνατότητα να κινηθείτε σε αυτήν την λίστα. Με τη χρήση δύο βέλων μπορείτε να εμφανίσετε τους επόμενους ή τους προηγούμενους ιστότοπους της κατάταξης. Επίσης μπορείτε να αναζητήσετε την θέση ενός ιστότοπου σε αυτή την λίστα, είτε στην κατάταξη για όλο το server, είτε ειδικά για κάποια κατηγορία.

Με μία περιήγηση στην ιστοσελίδα της υπηρεσίας μπορείτε να πληροφορηθείτε αναλυτικά για τις επισκέψεις που έχουν πραγματοποιηθεί καθώς και για το μέγεθος των δεδομένων που μεταφέρθη και αναλυτικά για το πότε έγιναν. Ακόμα μπορείτε να ενημερωθείτε για την κατανομή των επισκέψεων ανά ώρα αλλά και ανά κατηγορία. Οι παραπάνω πληροφορίες είναι διαθέσιμες για κάθε ιστότοπο, για κάθε κατηγορία και για το σύνολο του webserver.

5.3.2 Γραφήματα του graphs.php

Η παρακάτω εικόνα είναι γραφήματα που παράγονται δυναμικά από το αρχείο graphs.php χρησιμοποιώντας την βιβλιοθήκη Graphite.

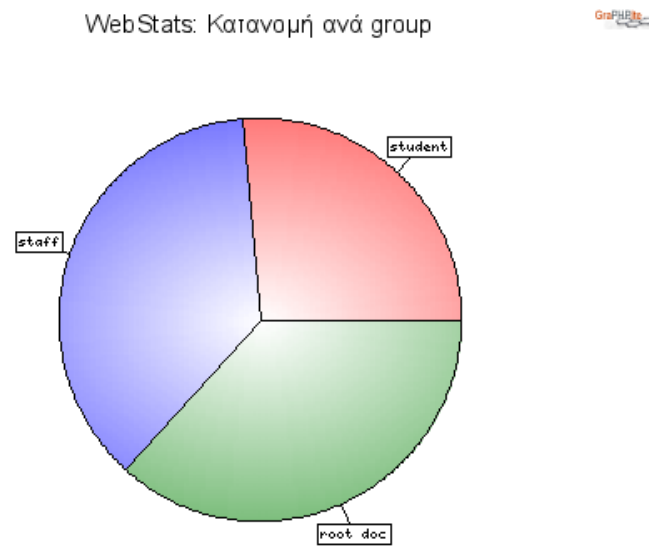


Figure 5.1: Η κατανομή των επισκέψεων ανά κατηγορία συνολικά

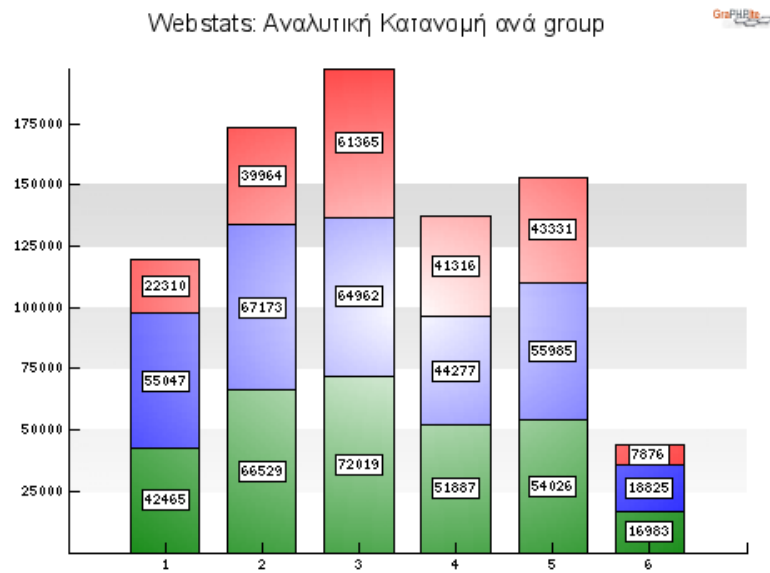


Figure 5.2: Η κατανομή των επισκέψεων ανά κατηγορία τους τελευταίους μήνες

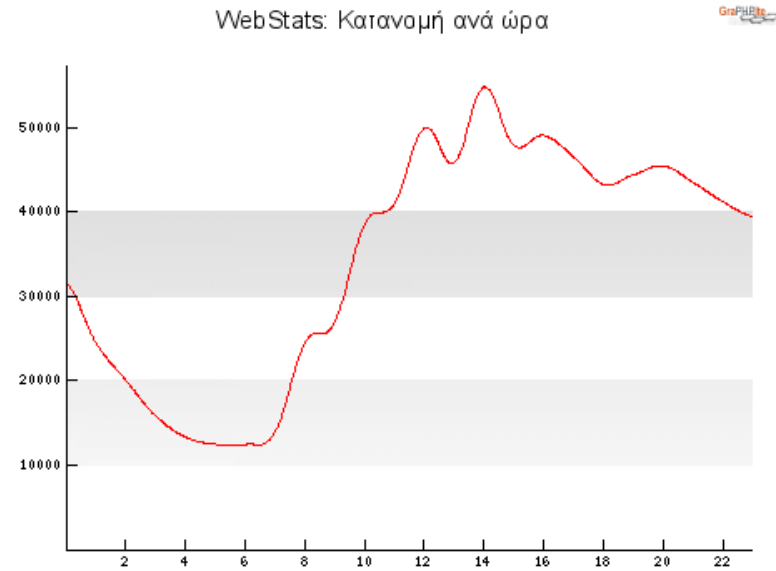


Figure 5.3: Η κατανομή των επισκέψεων ανά ώρα

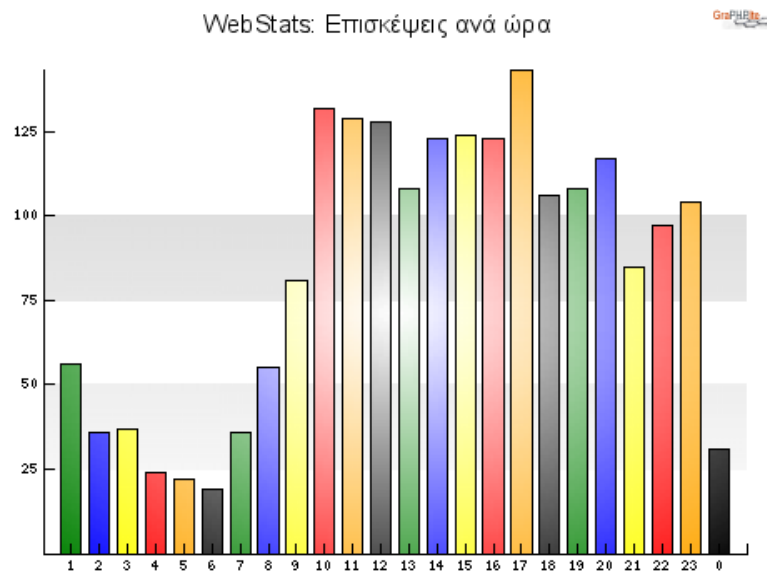


Figure 5.4: Γραφήμα στήλης για τις επισκέψεις των τελευταίων 24 ωρών

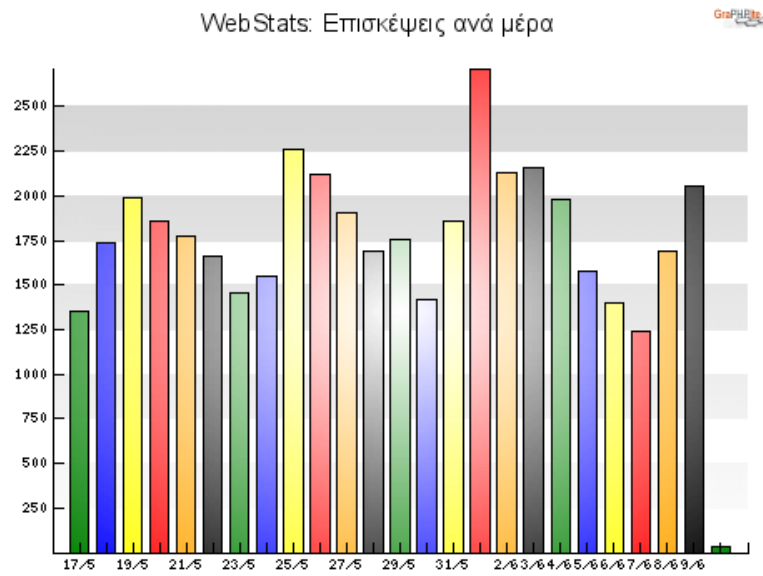


Figure 5.5: Γραφήμα στήλης για τις επισκέψεις των τελευταίων ημερών

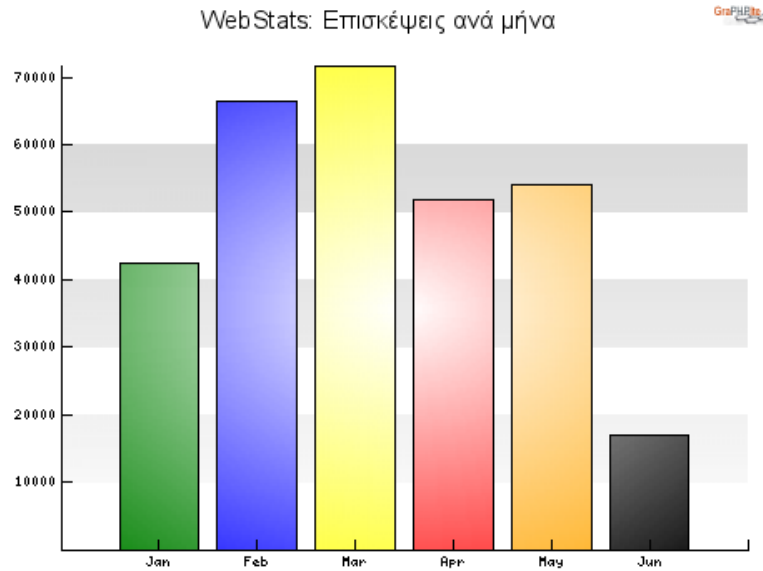


Figure 5.6: Γραφήμα στήλης για τις επισκέψεις των τελευταίων μηνών

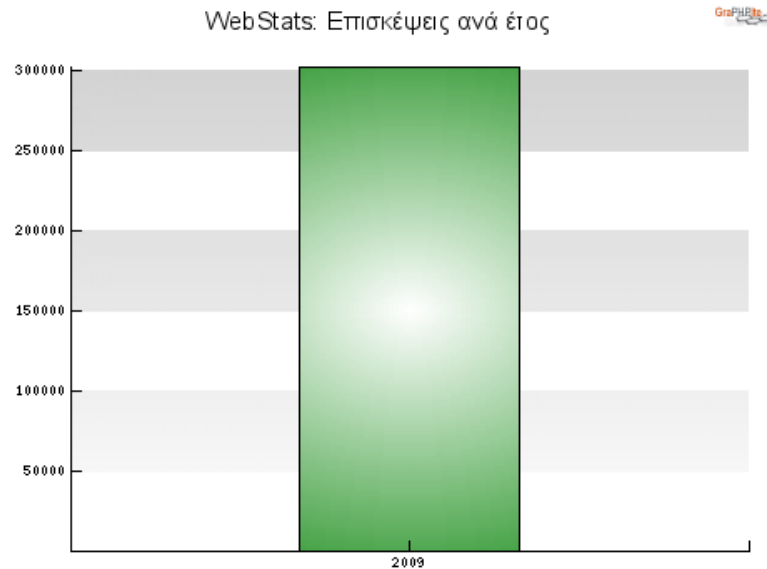


Figure 5.7: Γραφήμα στήλης για τις επισκέψεις ανά έτος

5.3.3 Counter

Το εργαλείο webstats προσφέρει την δυνατότητα στους χρήστες του aetos, τοποθέτησης counter στις ιστοσελίδες τους. Το αρχείο counter.php παίρνοντας σαν παράμετρο το όνομα ενός λογαριασμού υπολογίζει τον αριθμό των συνολικών επισκέψεων και όγκου δεδομένων, που έχουν κατέβει και με τη χρήση της βιβλιοθήκης GD2 τα εμφανίζει σε μία εικόνα. Εκτός των συνολικών επισκέψεων και δεδομένων εμφανίζει και από ποια ημερομηνία και έπειτα έχουν πραγματοποιηθεί αυτά τα στοιχεία. Η εικόνα που παράγεται είναι αρχείο τύπου png. Οδηγίες για την εγκατάσταση του counter δίνονται στην υποενότητα 6.2.1.

- Ο κώδικας του counter.php :

```

1 #!/usr/local/bin/php
2 <?php
3 function getcount($uname) {
4     //connect

```

```

5     $dbcon = pg_connect("dbname=$dbname user=$dbusr password=
        $passwd")
6     or die('Could not connect: ' . pg_last_error());
7
8     $result=pg_query("SELECT sum(h.hits) AS hits, sum(h.data)
        AS data FROM hpm h, usrdir u WHERE u.id=h.uid AND u.
        uname='$uname'")
9     or die('Query failed: ' . pg_last_error());
10    $hits = pg_fetch_result($result,0,0);
11    $data = pg_fetch_result($result,0,1);
12    //Free resultset
13    pg_free_result($result);
14
15    $result=pg_query("SELECT date_part('day',h.ts),date_part('
        month',h.ts),date_part('year',h.ts) FROM hpm h, usrdir u
        WHERE u.id=h.uid and u.uname='$uname' ORDER BY h.ts ASC
        LIMIT 1")
16    or die('Query failed: ' . pg_last_error());
17    $day = pg_fetch_result($result,0,0);
18    $month = pg_fetch_result($result,0,1);
19    $year = pg_fetch_result($result,0,2);
20    //Free resultset
21    pg_free_result($result);
22    $since_date = "$day-$month-$year";
23
24    $counter = array(1 => $hits, 2 => $data, 3 => $since_date);
25    return $counter;
26 } //eof getcount
27
28 $username = $_GET['username']; //onoma logariasmou
29 $counter = getcount($username);
30 $hits = $counter[1];
31 $data = $counter[2];
32 $since_date = $counter[3];
33 $title = " webstats: $username ";
34 $bottomline = " since $since_date ";
35 $font = 5;
36 $font2 = 3;
37 $stringhits = " $hits hits ";
38
39 if($data > 1024 ) {

```



```
40     $data = $data/1024;
41     $data = round($data,2);
42     $stringdata = " $data MB "; }
43 else {
44     $stringdata = " $data KB "; }
45
46 if(strlen($stringhits) > strlen($stringdata) and strlen(
    $stringhits) > strlen($title) and strlen($stringhits) >
    strlen($bottomline) ) {
47     $width = ImageFontWidth($font) * strlen($stringhits); }
48 elseif (strlen($stringdata) > strlen($title) and strlen(
    $stringdata) > strlen($bottomline) ) {
49     $width = ImageFontWidth($font) * strlen($stringdata); }
50 else {
51     if(strlen($bottomline) > strlen($title) ) {
52         $width = ImageFontWidth($font2) * strlen($bottomline);
53     }
54     else {
55         $width = ImageFontWidth($font2) * strlen($title); }
56 }
57 $height = ImageFontHeight($font) + ImageFontHeight($font2) +
    10;
58 $im = imagecreatetruecolor($width+4, $height+21);
59 $white = imagecolorallocate($im, 255, 255, 255);
60 $black = imagecolorallocate($im, 0, 0,0);
61 imagefilledrectangle($im, 3, 14, $width, $height+8, $white);
62 imagestring ($im, $font2, 4, 1, $title, $white);
63 imagestring ($im, $font, 4, 15, $stringhits, $black);
64 imagestring ($im, $font, 4, $height/2+11, $stringdata, $black)
    ;
65 imagestring ($im, $font2, 4, $height+8, $bottomline, $white);
66 header ("Content-type: image/png");
67 imagepng ($im);
68 imagedestroy ($im);
69 ?>
```


6 ΕΓΚΑΤΑΣΤΑΣΗ

6.1 ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ

1. Logtail
2. Apache Web Server, όπου τα access logs θα είναι σε common ή combined μορφή
3. Python 2.x
4. pyPgSQL
5. PostgreSQL 7.x ή νεότερη έκδοση
6. PHP 4 ή νεότερη έκδοση
7. PHP PostgreSQL Driver
8. Η βιβλιοθήκη GD 2 της PHP
9. Smarty Template Engine

6.2 ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ

Εφόσον πληρούνται όλες οι απαιτήσεις τότε απομένουν μερικά απλά βήματα ώστε η υπηρεσία webstats να τρέξει σε ένα σύστημα. Θα πρέπει να τονιστεί πως το όλο εγχείρημα είναι χτισμένο για τον server του τμήματος πληροφορικής, μόνο σε αυτόν μπορεί να λειτουργήσει. Βέβαια μετά από αρκετές μετατροπές θα ήταν δυνατό να τρέξει και σε άλλο μηχάνημα.

Η ουσιαστική αλλαγή που θα πρέπει να γίνει για την μεταφορά αυτής της εφαρμογής σε ένα άλλο μηχάνημα είναι ο τρόπος κατηγοριοποίησης των λογαριασμών του server. Στο server aetos οι λογαριασμοί χωρίζονται σε staff και student, δηλαδή αυτούς που ανήκουν στο προσωπικό του τμήματος όπως οι καθηγητές και αυτούς που ανήκουν σε φοιτητές. Οι μετατροπές που θα πρέπει να γίνουν είναι στην συνάρτηση getGroup του αρχείου parser.py αλλά και στο constraint που διασφαλίζει τις σωστές τιμές για το πεδίο ugroup του πίνακα usrdir της βάσης. Το αρχείο με τον sql κώδικα για την δημιουργία των πινάκων είναι το webstats_tables.sql.

Αυτό που θα πρέπει να αλλάξει στην συνάρτηση είναι η λογική με την οποία κατηγοριοποιούμε τους λογαριασμούς. Για παράδειγμα στην συνάρτηση getGroup του parser εξετάζεται το home directory ενός λογαριασμού και κατηγοριοποιείται ανάλογα. Αν το path ξεκινάει

με `/home/staff` το `group` παίρνει την τιμή `staff`, αν ξεκινάει με `/home/student` παίρνει τιμή `student`, αλλιώς παίρνει τιμή `other`. Αυτή η λογική βασίστηκε στον τρόπο που οργανώνονται οι λογαριασμοί στον `server` του τμήματος. Ο τρόπος οργάνωσης των λογαριασμών είναι αυτό που καθορίζει την λειτουργία της συνάρτησης `getGroup`.

Όπως αναφέραμε και στην προηγούμενη υποενότητα, στις απαιτήσεις του συστήματος, οι εγγραφές του `apache` για τα `access logs` θα πρέπει να είναι σε μορφή `combined`.

Παρακάτω αναφέρονται οι οδηγίες της εγκατάστασης βήμα προς βήμα.

1. Αντιγραφή των αρχείων `launcher.sh` και `python.py` στο σύστημα μας.
2. Επεξεργασία του αρχείου `launcher.sh`, δήλωση των `full paths` των αρχείων που καλούνται.
3. Δημιουργία των πινάκων της βάσης στην `Postgresql`. Οι εντολές για την δημιουργία τους βρίσκονται στο αρχείο `webstats_tables.sql`.
4. Επεξεργασία του αρχείου `parser.py`. Δήλωση των τιμών των μεταβλητών `database` και `password`, το όνομα της βάσης όπου δημιουργήθηκαν οι πίνακες και το κωδικό πρόσβασης σε αυτήν αντίστοιχα.
5. Επόμενο βήμα είναι η προσθήκη του `web interface`. Αντιγραφή των αρχείων στο σύστημα και δήλωση τιμών στα `php` αρχεία για το όνομα και το `password` της βάσης.
6. Δημιουργία ενός φακέλου με το όνομα `smarty` και 4 ακόμα υποφακέλους με τα ονόματα:
 - `templates`
 - `templates_c`
 - `cache`
 - `configs`
7. Αντιγραφή των `smarty templates`, τα αρχεία με κατάληξη `tpl`, στον φάκελο που δημιουργήθηκε πριν με όνομα `templates`.
8. Τέλος η δημιουργία ένα δαίμονα που θα εκτελεί κάθε 15 λεπτά το αρχείο `parser.py`. Στο κεφάλαιο `ΧΡΗΣΗ` δίνονται οδηγίες για το πως γίνεται.

Αυτό που θα πρέπει να προσεχθεί είναι τα δικαιώματα πρόσβασης που θα ορισθούν για τα αρχεία της εφαρμογής. Ειδικότερα για τα αρχεία που συνδέονται στην βάση και περιέχουν το κωδικό πρόσβασης συνιστάται η αφαίρεση του δικαιώματος ανάγνωσης και εγγραφής σε αυτά. Επίσης στα αρχεία της ιστοσελίδας της υπηρεσίας να δωθούν δικαιώματα ανάγνωσης ώστε να είναι δυνατή η εμφάνισή τους από τους `web browsers`.

6.2.1 Εγκατάσταση counter σε ιστοσελίδα

Για την εμφάνιση του counter σε ιστοσελίδα θα πρέπει να δημιουργηθεί ένα image tag και στην ιδιότητα src να δωθεί η τιμή της διαδρομής του αρχείου counter.php καθώς και να δηλωθεί το όνομα του λογαριασμού του οποίου τα στοιχεία θα εμφανισθούν. Για παράδειγμα, εάν πρόκειται να εμφανίσουμε τον αριθμό επισκέψεων του ιστότοπου του λογαριασμού apnav θα πρέπει προστεθεί ο παρακάτω κώδικας στην εν λόγω ιστοσελίδα.

```

```

Ο counter έχει τη μορφή :

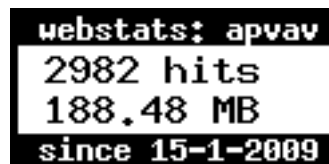


Figure 6.1: counter

7 ΧΡΗΣΗ

Το webstats ξεκίνησε να συλλέγει δεδομένα για την επισκεψιμότητα στον web server του τμήματος από τις 15 Ιανουαρίου του 2009. Παρακάτω παρατίθεται ένας πίνακας με στοιχεία για την λειτουργία του εργαλείου για ένα διάστημα λίγο μεγαλύτερο των πέντε μηνών.

| Όνομα Πίνακα | Αριθμός Εγγραφών |
|--------------|------------------|
| usrdir | 289 |
| hph | 933 |
| hpd | 3820 |
| hpm | 1521 |
| hhm | 22099 |

| | |
|----------------------------|-------|
| Συνολικός Αριθμός Εγγραφών | 28662 |
|----------------------------|-------|

7.1 SHELL ΕΝΤΟΛΕΣ

- Για την δημιουργία της βάσης αφού συνδεθούμε στην Postgresql, δίνεται η εντολή σε psql περιβάλλον.

```
CREATE DATABASE dbname OWNER username;
```

- Για την δημιουργία των πινακων δίνετε η παρακάτω εντολή το shell.

```
psq -d dbname -U username -f webstats_tables.sql
```

- Η μορφή της εντολής logtail. Το filename είναι το αρχείο προς ανάγνωση και το offsetfile το αρχείο που έχει τις offset εγγραφές.

```
logtail -ffilename -ooffsetfile
```

- Για την εκτέλεση του parser δίνουμε σαν δεύτερη παράμετρο το αρχείο με τα logs.

```
parser.py filename
```

- Για την δημιουργία ενός δαίμονα για την εκτελεση του launcher.ssh και την σάρωση των access logs του webserver και την ενημέρωση της βάσης ανά τακτά διαστήματα θα χρησιμοποιήθει το Cron. Ένα εργαλείο για την δήλωση προγραμματισμένων εργασιών. Η εντολή που θα πρέπει να δωθεί είναι η παρακάτω.

```
crontab -u user -e
```

- Αφού ανοιχθεί το αρχείο με τις προγραμματισμένες εργασίες, εισάγετε την παρακάτω γραμμή και σώζετε το αρχείο. Θα πρέπει να δηλώθει όλη η διαδρομή για την εργασία που θα εκτελείται. Η παρακάτω εγγραφή σημαίνει ότι στο 15ο, 30ο, 45ο και 59ο λεπτό για κάθε ώρα, κάθε μήνα και κάθε χρόνο θα εκτελείται το script launcher.sh, δηλαδή κάθε δεκαπέντε λεπτά. Το output της εντολής διοχετεύεται στο /dev/null ώστε να μην αποστέλεται κάθε φορά με e-mail στον χρήστη που το εκτελεί.

```
15,30,45,59 * * * * launcher.sh > /dev/null
```

7.2 ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ

Σε αυτήν την υποενότητα δίνονται οδηγίες για το πως μπορείτε να εμφανίσετε αποτελέσματα στην ιστοσελίδα. Για μεγαλύτερη κατανόηση γίνεται χρήση εικόνων.

- Για την εμφάνιση στοιχείων για το server ή για κάποια κατηγορία ή για ένα χρήστη, πρέπει να επιλεγεί από το κεντρικό μενού η “Αναζήτηση”. Στη συνέχεια ή εκτελούμε μία αναζήτηση για κάποιο χρήστη ή εμφανίζουμε μία εκ των επιλογών στο υπομενού αριστερά. Η επιλογές είναι :

1. server
2. Κεντρική Σελίδα
3. Προσωπικό
4. Φοιτητές

Figure 7.1: Αναζήτηση

- Η παρακάτω εικόνα είναι η εμφάνιση στοιχείων για την κεντρική σελίδα. Επιπλέον μπορούμε να εμφανίσουμε την κατανομή ανά ώρα ή τις περιόδους με τις περισσότερες επισκέψεις για αυτήν την σελίδα από το υπομενού αριστερά. Επίσης στο κάτω μέρος της σελίδας βρίσκονται links για την εμφάνιση γραφημάτων για αναλυτική παρουσίαση της επισκεψιμότητας.

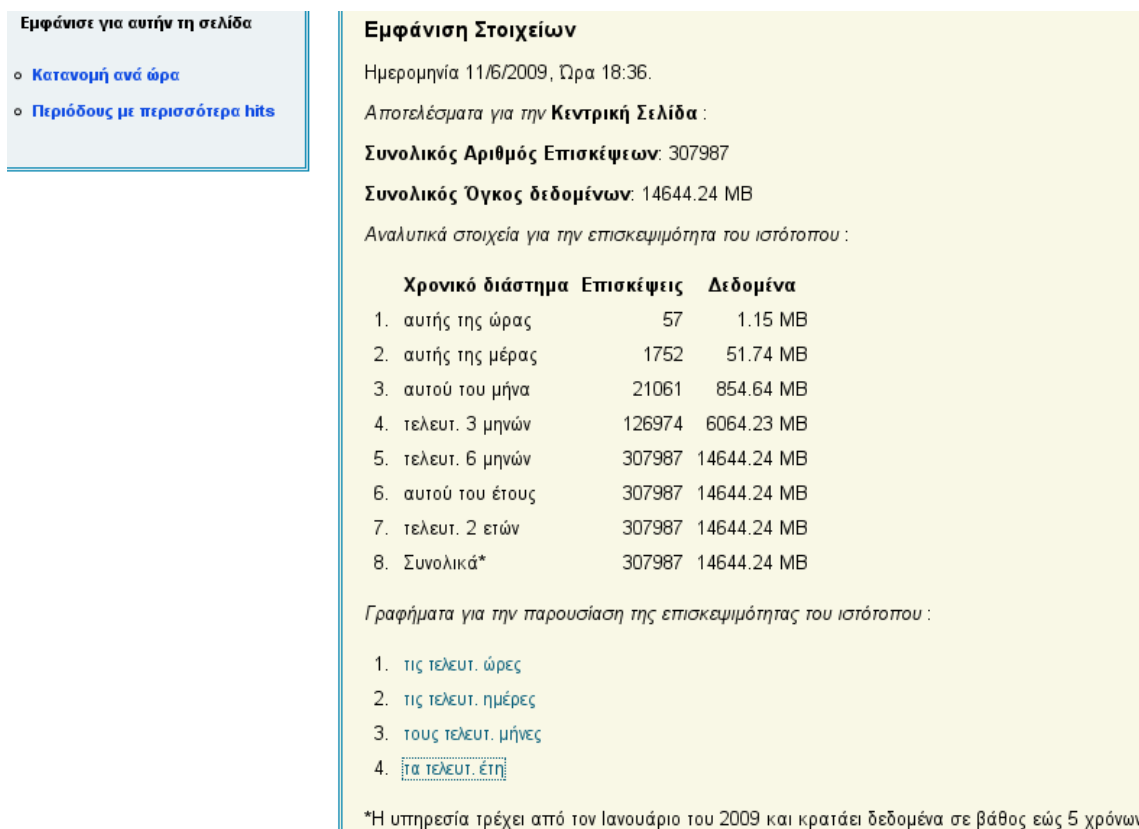


Figure 7.2: Εμφάνιση Στοιχείων

- Έχοντας εμφανίσει το γράφημα για την αναπαράσταση της κίνησης για ένα από τα χρονικά διαστήματα που έχουν αναφερθεί, υπάρχει η δυνατότητα εμφάνισης των υπολοίπων από το υπομενού αριστερά. Όσον αφορά το διάστημα των τελευταίων ετών υπάρχουν κάποιες περαιτέρω επιλογές που αφορούν την αναλυτική παρουσίαση της κίνησης για κάθε έτος ειδικά. Στην εικόνα που ακολουθεί εμφανίζεται η κίνηση των τελευταίων μηνών :

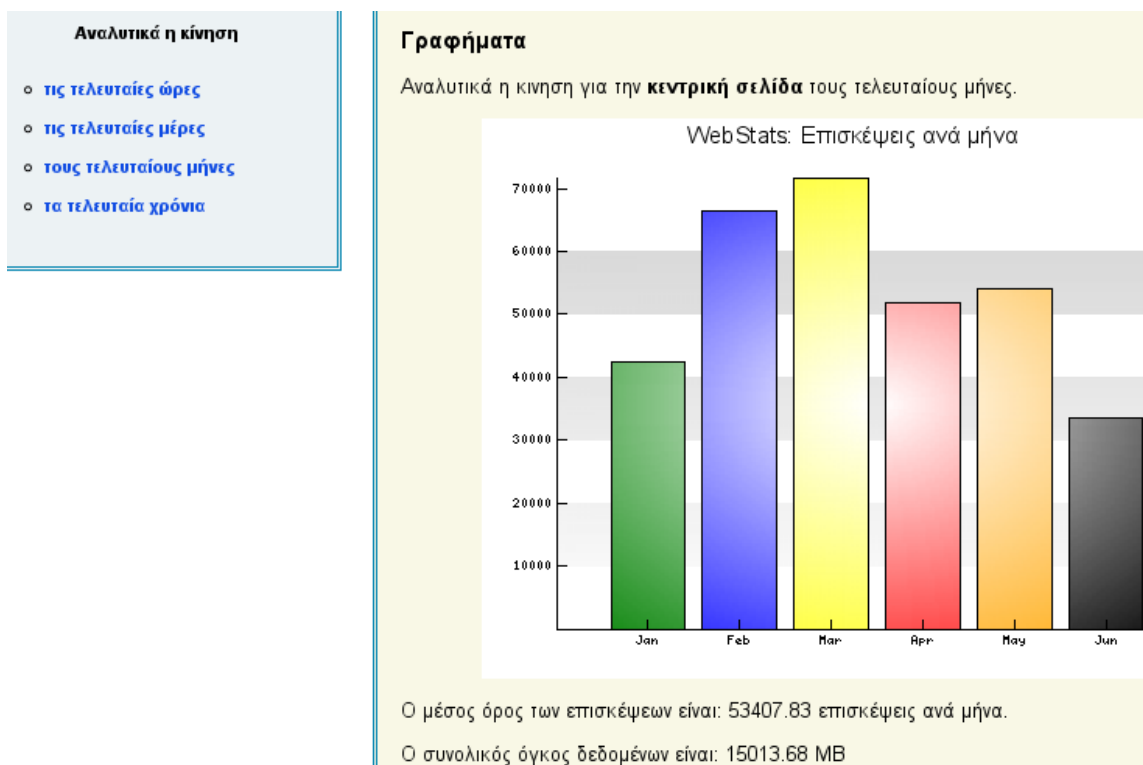


Figure 7.3: Γράφημα αριθμ. επισκέψεων των τελευταίους μηνών

- Αν επιλέξουμε “Κατανομή ανά ώρα” από το υπομενού στην σελίδα “Εμφάνιση Στοιχείων”, υπάρχει η δυνατότητα της εμφάνισης αυτής της κατανομής για περισσότερα από ένα χρονικά διαστήματα. Αυτά τα διαστήματα είναι των τελευταίων δώδεκα έως ενός μηνών. Η επιλογή γίνεται από την λίστα στο κάτω μέρος της σελίδας. Δείτε στην εικόνα :

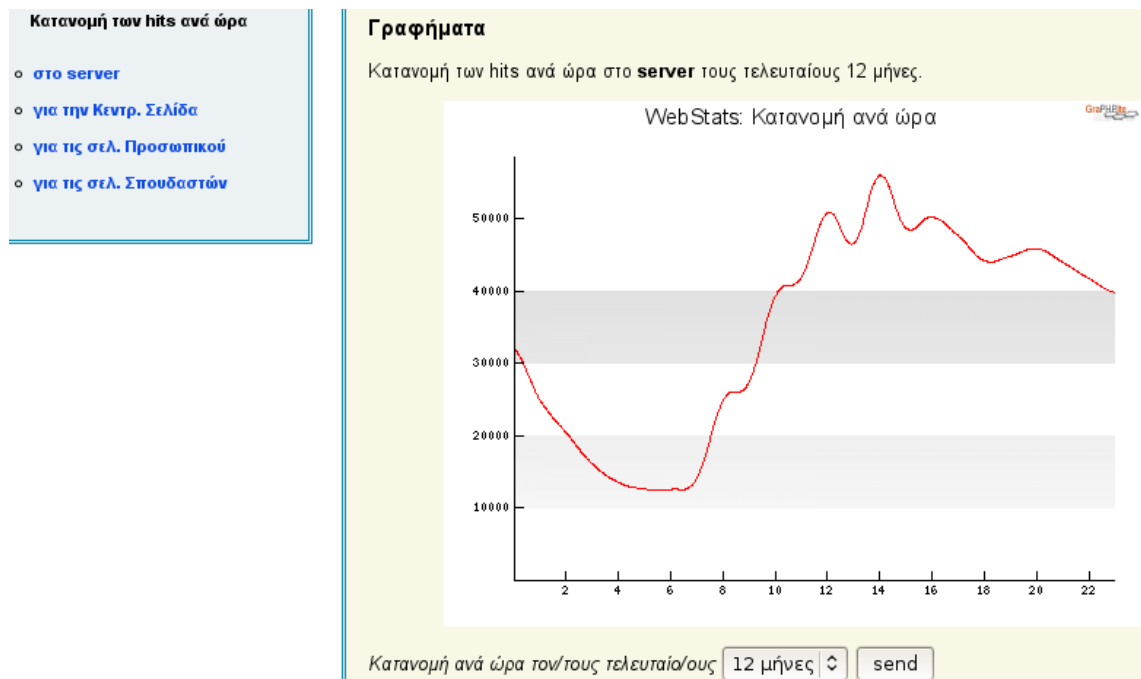


Figure 7.4: Κατανομή ανά ώρα

- Για να εμφανίσουμε τα στατιστικά της σημερινής ημέρας, επιλέγουμε “Στατιστικά” από το κεντρικό μενού. Στην σελίδα Στατιστικά, στο υπομενού αριστερά υπάρχουν οι επιλογές :

1. Δημοφιλέστερες Σελίδες
2. Περίοδοι με τα περισσότερα hits (στο server)
3. Κατανομή ανά group
4. Κατανομή ανά ώρα (στο server)

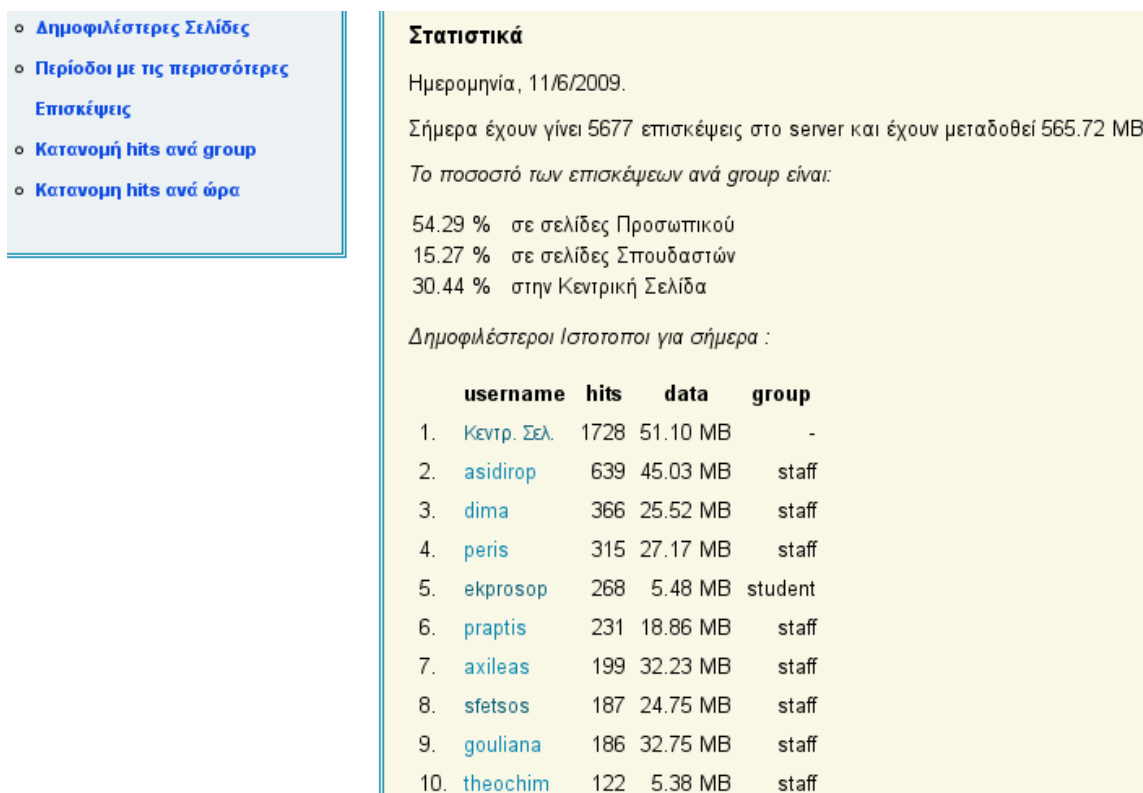


Figure 7.5: Στατιστικά

- Εάν επιλέξουμε “Δημοφιλέστερες Σελίδες” από την σελίδα “Στατιστικά”, μπορούμε να εμφανίσουμε τις δημοφιλέστερες σελίδες σε όλο τον server ή και για κάθε κατηγορία ξεχωριστά, αυτές οι επιλογές βρίσκονται στο υπομενού αριστερά. Από την φόρμα στο κάτω μέρος της σελίδας μπορούμε να αναζητήσουμε την θέση ενός χρήστη σε αυτήν την λίστα, είτε για όλο τον server, είτε για το group που ανήκει. Ακόμα υπάρχουν δύο βέλη για να εμφανίσουμε τις επόμενες και τις προηγούμενες θέσεις. Δείτε στην εικόνα :

Πιο δημοφιλείς σελίδες

- στο **server**
- στις **σελίδες προσωπικού**
- στις **σελίδες σπουδαστών**

Δημοφιλέστερες Σελίδες

Σελίδες με την μεγαλύτερη επισκεψιμότητα στο **server**.

| | username | hits | data | group |
|-----|-------------|--------|-------------|---------|
| 1. | Κεντρ. Σελ. | 307963 | 14643.55 MB | - |
| 2. | ekprosop | 86738 | 1619.69 MB | student |
| 3. | kdiamant | 29371 | 10311.72 MB | staff |
| 4. | asidirop | 28776 | 3239.69 MB | staff |
| 5. | peris | 26617 | 3042.10 MB | staff |
| 6. | theochim | 23453 | 2709.65 MB | staff |
| 7. | free | 23438 | 1815.71 MB | staff |
| 8. | axileas | 18232 | 4581.72 MB | staff |
| 9. | dima | 17934 | 2033.86 MB | staff |
| 10. | sfetsos | 15738 | 2614.82 MB | staff |

<< >>

Αναζητήστε τη θέση ενός λογαριασμού.

username:

Figure 7.6: Δημοφιλέστερες Σελίδες

- Την κατανομή των επισκέψεων ανά group μπορούμε να την επιλέξουμε από το υπομενού της σελίδας “Στατιστικά”. Στο κάτω μέρος της σελίδας υπάρχει μια λίστα για τον υπολογισμό της κατανομής για τα εξής χρονικά διαστήματα :
 1. αυτόν το μήνα
 2. τους τελευταίους τρεις μήνες
 3. τους τελευταίους έξι μήνες
 4. αυτό το έτος
 5. τα τελευταία δύο έτη
 6. Συνολικά

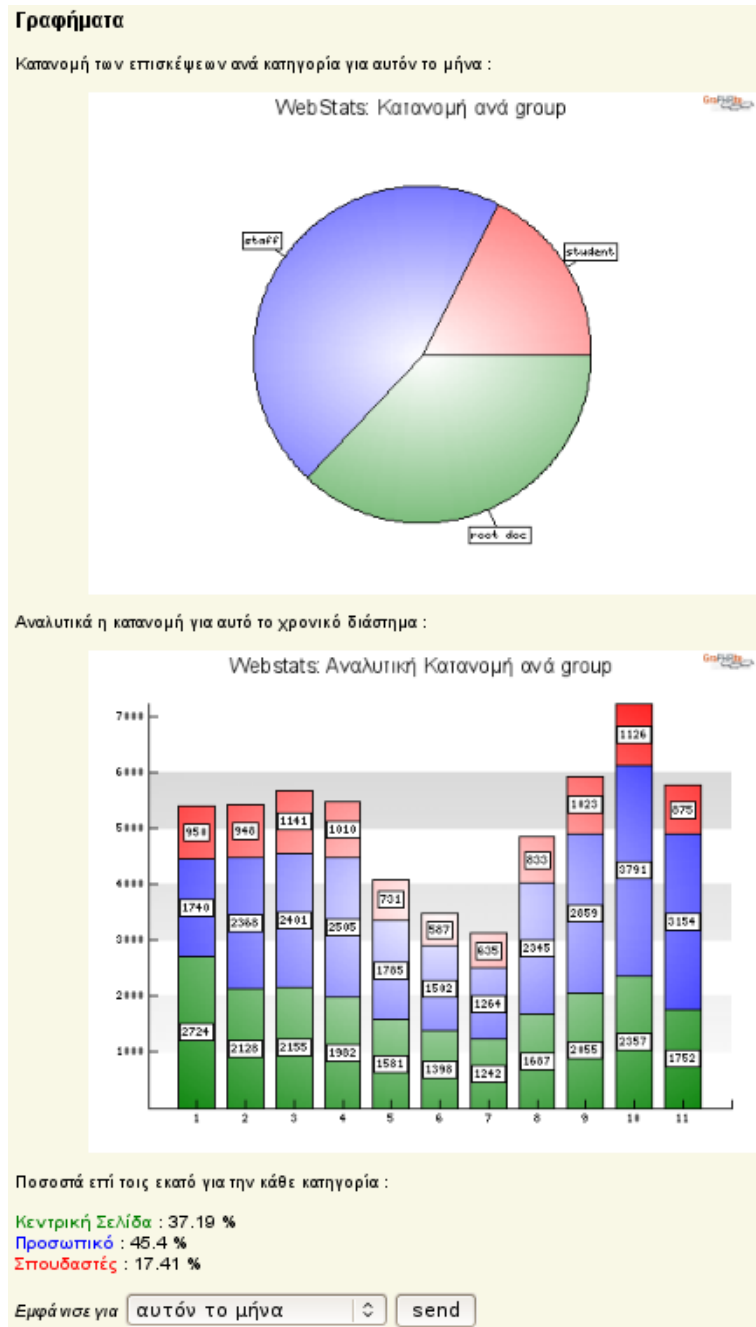


Figure 7.7: Κατανομή ανά group

8 ΣΧΟΛΙΑ / ΠΑΡΑΤΗΡΗΣΕΙΣ

Η υπηρεσία webstats είναι κατάλληλη για ένα webserver στον οποίο το module userdir του apache είναι ενεργοποιημένο έτσι ώστε να μπορεί ο κάθε λογαριασμός στον server να ανεβάσει την δική του ιστοσελίδα, όπως για παράδειγμα στον server aetos του τμήματος πληροφορικής. Αυτό το εργαλείο δίνει την δυνατότητα της ενημέρωσης για την επισκεψιμότητα της κάθε ιστοσελίδας, για τις κατηγορίες των φοιτητών και των καθηγητών αλλά και για όλον το server.

Για την υλοποίηση αυτής της εφαρμογής χρησιμοποιήθηκαν εργαλεία που διατίθενται ελεύθερα όπως η rython για τον parser, η postgres για την βάση και η rhr για το web interface καθώς και κάποιες περαιτέρω βιβλιοθήκες. Η εφαρμογή αναπτύχθηκε σε περιβάλλον Unix.

Η μεταφορά του σε ένα άλλο σύστημα θα είναι δυνατή εφόσον πληρούνται όλες οι απαιτήσεις συστήματος όμως θα πρέπει να γίνουν κάποιες σημαντικές μετατροπές. Αυτές οφείλονται κυρίως στα κριτήρια με τα οποία ομαδοποιούνται οι λογαριασμοί στον εκάστοτε web server. Αυτό σημαίνει ότι θα πρέπει να γίνουν αλλαγές στο κώδικα του parser στη συνάρτηση που ομαδοποιεί τους λογαριασμούς καθώς και στο constraint του πεδίου ugroup του πίνακα usrdir που εξασφαλίζει αποδεκτές τιμές για τις εγγραφές. Ακόμα θα πρέπει να γίνουν αλλαγές στο περιεχόμενο της ιστοσελίδας και στα rhr αρχεία που εκτελούν ερωτήματα στη βάση.

Όσον αφορά την μεταφορά σε άλλο σύστημα όπως για παράδειγμα το Windows Server, αυτό είναι δυνατόν με κάποιες ακόμα αλλαγές. Αυτές είναι το shell script launcher.sh για την εκτέλεση του parser, η χρήση ενός προγράμματος παρόμοιου της εντολής logtail που χρησιμοποιείται για την ανάγνωση νέων εγγραφών σε κάποιο αρχείο και η χρήση ενός scheduler για προγραμματισμένες εκτελέσεις παρόμοιο με το Cron. Επίσης είναι απαραίτητες κάποιες αλλαγές στον κώδικα του parser όσον αφορά την εξόρυξη πληροφοριών για τους λογαριασμούς του συστήματος.

9 ΕΠΙΛΟΓΟΣ

Η υπηρεσία webstats μας ενημερώνει με στατιστικά στοιχεία σχετικά με την web κίνηση και είναι χτισμένο στην λογική του server του τμήματος πληροφορικής. Είναι ένα εργαλείο το οποίο δουλεύει όχι μόνο για το σύνολο ενός web server αλλά και για κάθε λογαριασμό ξεχωριστά, σε αντίθεση με τα παρόμοια υπάρχοντα εργαλεία. Η μετατροπή του για ένα άλλο μηχάνημα είναι μία απαιτητική διαδικασία και έχει αναλυθεί σε προηγούμενη παράγραφο.

Η υλοποίηση αυτής της εφαρμογής απαιτήσε γνώσεις προγραμματισμού για την ανάπτυξη του parser, γνώσεις βάσεων δεδομένων για τον σχεδιασμό της βάσης και γνώσεις διαδικτυακού προγραμματισμού για την υλοποίηση του web interface. Απαραίτητες κρίθηκαν και στοιχειώδεις γνώσεις Unix συστημάτων για την εγκατάσταση και χρήση του εργαλείου.

Bibliography

- [1] Mark Lutz, O'Reilly Learning Python, 3rd Edition, *O'Reilly Media, Inc*, 2008
- [2] Raghuram Ramakrishnan, Johannes Gehrke, Συστήματα Διαχείρισης Βάσεων Δεδομένων Τόμος Α, 2η Έκδοση, *Εκδόσεις Τζιολα*, 2001
- [3] Python Online Documentation, '<http://www.python.org/doc/>', *Python Software Foundation*, 2009
- [4] PostgreSQL Online Documentation, '<http://www.postgresql.org/docs/>', *PostgreSQL Global Development Group*, 2008
- [5] PHP Online Documentation, '<http://www.php.net/manual/en/>', *The PHP Group*, 2009
- [6] Smarty Templates Engine Online Documentation, '<http://www.smarty.net/manual/en/>', *The PHP Group*, 2007
- [7] Graphpate Online Documentation, '<http://graphpate.sourceforge.net/docs/>', *phpDocumentor 1.2.3*, 2004
- [8] Apache Online Documentation, '<http://httpd.apache.org/docs/>', *The Apache Software Foundation*, 2009
- [9] HTML 4.01 Specification, '<http://www.w3.org/TR/REC-html40/>', *W3C*, 2009
- [10] CSS 2.1 Specification, '<http://www.w3.org/TR/CSS21/>', *W3C*, 2009
- [11] Awstats Online Documentation, '<http://awstats.sourceforge.net/docs/index.html>'
- [12] Analog Online Documentation, '<http://analog.cx/docs/Readme.html>'
- [13] Webalizer Guide, '<http://www.mrunix.net/webalizer/simpleton.html>'