

Odyssey track



TIP 2006

Medawar Elias
elias.medawar@ltsi.ch

Supérieur professionnel :

Hirtel Jacques
032 942 43 15
LTSI
jacques@hirtzel.ch

Experts désignés :

Gagnebin Pascal
--
CPAIJB
pascal.gagnebin@cpaijb.ch

Girard Alain
032 942 54 80
LONGINES
alain.girard@longines.com

Table des matières :

TABLE DES MATIERES :	2
1. INTRODUCTION :	3
1.1. ETAT DE DEPART :	3
1.2. OBJECTIF 1 :	3
1.3. OBJECTIF 2 :	4
1.4. UTILITE DU TRAVAIL :	4
1.5. MOYENS A DISPOSITION :	4
2. STRUCTURE.	5
2.1. STRUCTURE DES DOSSIERS:.....	5
2.2. STRUCTURE DE L'APPLICATION :	6
2.3. DEROULEMENT DE LA PROCEDURE DE GENERATION DU GRAPHIQUE :	7
3. DETAIL DU TRAVAIL	7
3.1. ETUDE DES TECHNOLOGIES	7
3.2. LA LIBRAIRIE GD.....	7
3.3. NOMBRE DE VISITES.....	8
3.4. NOMBRE DE CLICS.	10
3.5. CHEMINS EMPRUNTE :	11
3.6. GENERATION DU GRAPHIQUE :	13
3.7. GENERATION DU CUBE OLAP :	15
4. CONCLUSION :	20

1. Introduction :

1.1. Etat de départ :

Les logs des sites utilisant Odyssey, sont enregistrés dans la base de données mais il n’y a aucun affichage de ces données.

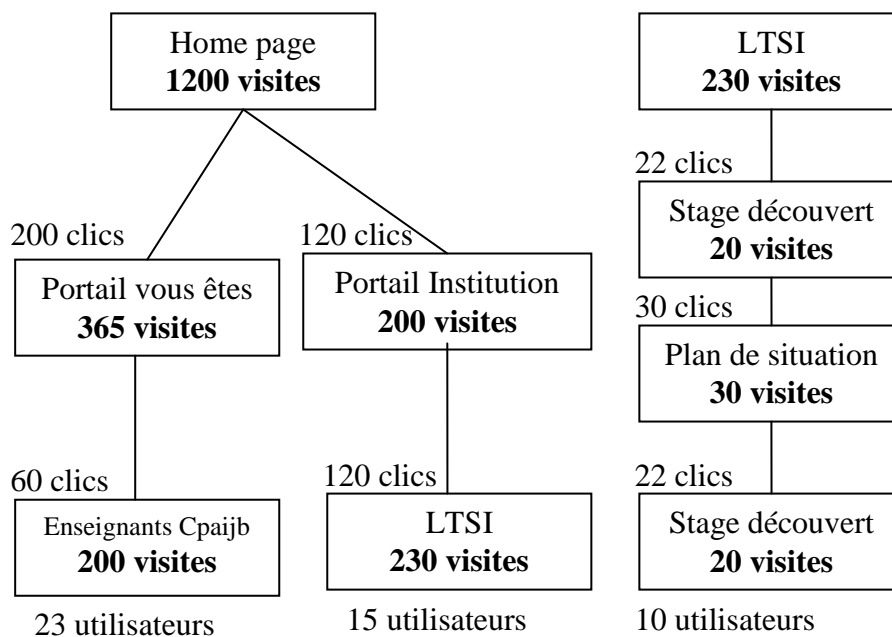
Table t_stat.

Champ	Type	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> id	int(10)		Non		auto_increment	
<input type="checkbox"/> res_stat	tinytext		Non			
<input type="checkbox"/> ip_stat	varchar(15)		Non			
<input type="checkbox"/> dom_stat	tinytext		Non			
<input type="checkbox"/> browser_stat	tinytext		Non			
<input type="checkbox"/> ref_stat	varchar(200)		Non			
<input type="checkbox"/> ref_page	varchar(200)		Non			
<input type="checkbox"/> date_stat	datetime		Non	0000-00-00 00:00:00		
<input type="checkbox"/> os_stat	varchar(20)		Non			

1.2. Objectif 1 :

Il est demandé dans le cadre du travail individuel productif de faire une application qui représente les chemins les plus empruntés sur le site.

Exemple :



— = Chemin parcouru

Définition de jargon :

Afin de clarifier les expressions et la situation, voici les explications de ce qui à été compris et définit avec M. Hirtzel.

Visite : Affichage d'une page par un utilisateur. Si cette opération est répétée dans un intervalle de 20 minutes, cet affichage ne compte que pour une seule visite.

Utilisateur : Personne ayant une même adresse IP durant la visite du site internet. Dans le cas où plusieurs utilisateurs ont la même adresse IP (par exemple en passant par un Proxy), ils sont ignorés.

Chemin parcouru : Un chemin débute à la première page de la visite et se termine quand l'utilisateur n'est plus sur le site ou que l'intervalle entre deux visites de page est plus grand que 20 minutes (ce qui crée un autre chemin). Si l'utilisateur repasse sur son chemin, on ne traite pas le cas.

Clic : Un clic est l'action qui relie deux pages. Le temps entre les clics, n'est pas pris en compte au contraire d'une visite. Donc si l'on clics 10 fois en 1 minute entre une page et une autre on compte 10fois.

1.3. Objectif 2 :

IL est demandé de générer un cube OLAP qui contient les informations utiles pour les statistiques.

1.4. Utilité du travail :

Grâce à cet outil, nous pourrons voir ce que font les gens sur notre site. Ces informations sont précieuses pour améliorer la structure ainsi que le contenu de notre site.

1.5. Moyens à disposition :

- Langage de programmation PHP , MYSQL,HTML
- Serveur web www.cpaijb.ch

Caractéristique :

Processeur : Intel Pentium 3 XEON 797 MHz

Ram : 640 MB

Os : Windows Server 2003

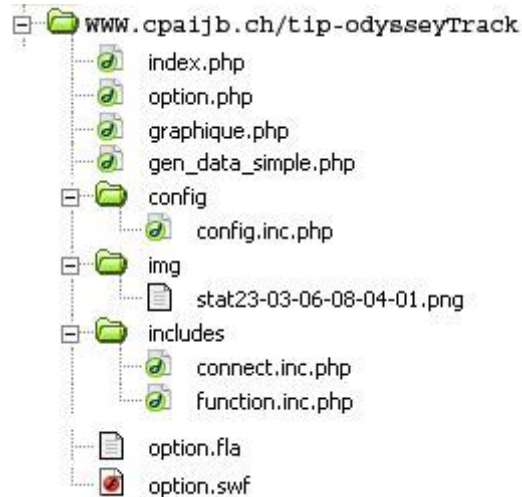
PHP : version 4.3.9

Mysql : 4.0.21

Droit d'accès : Administrateur.

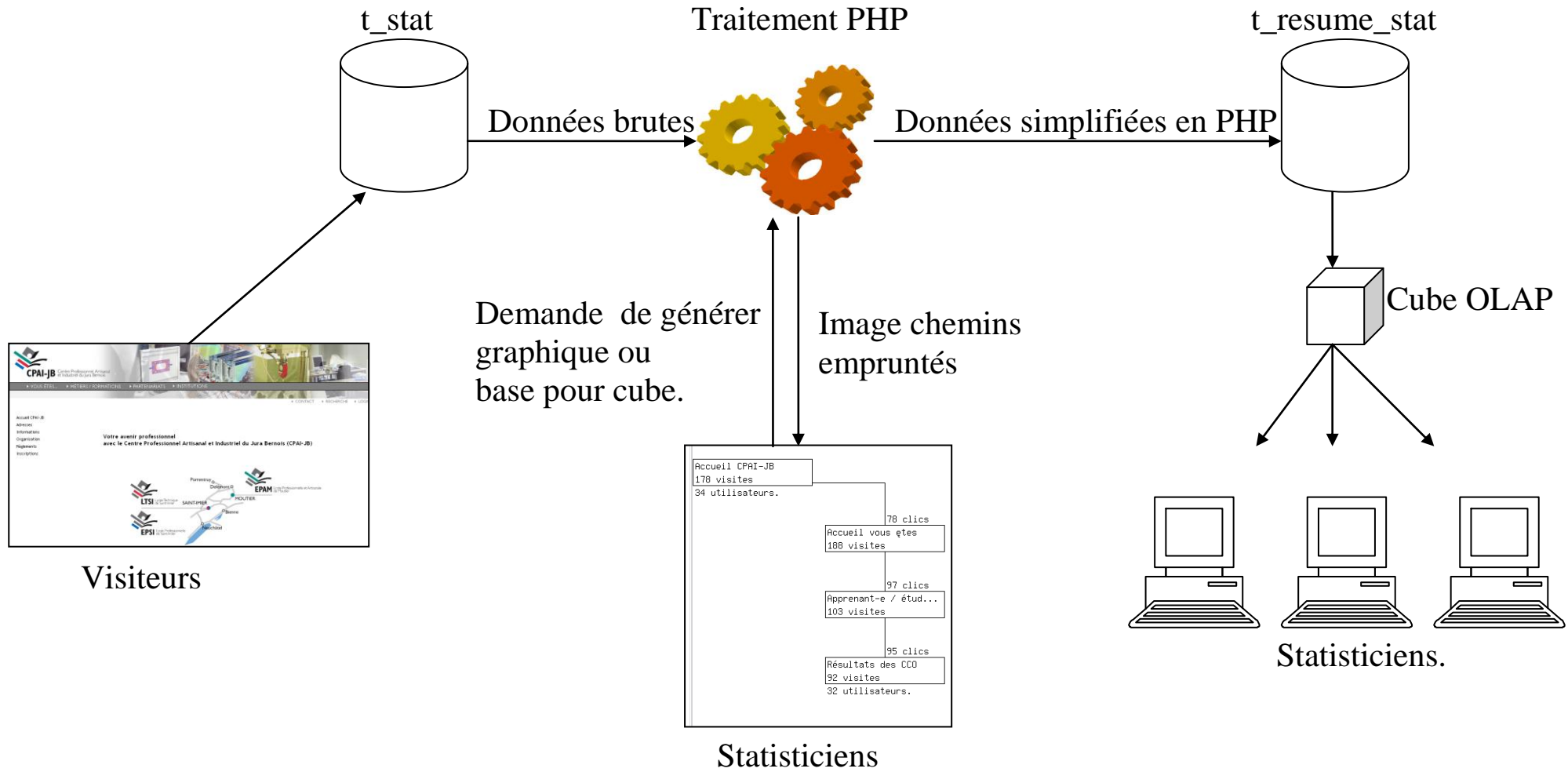
2. Structure.

2.1. Structure des dossiers:

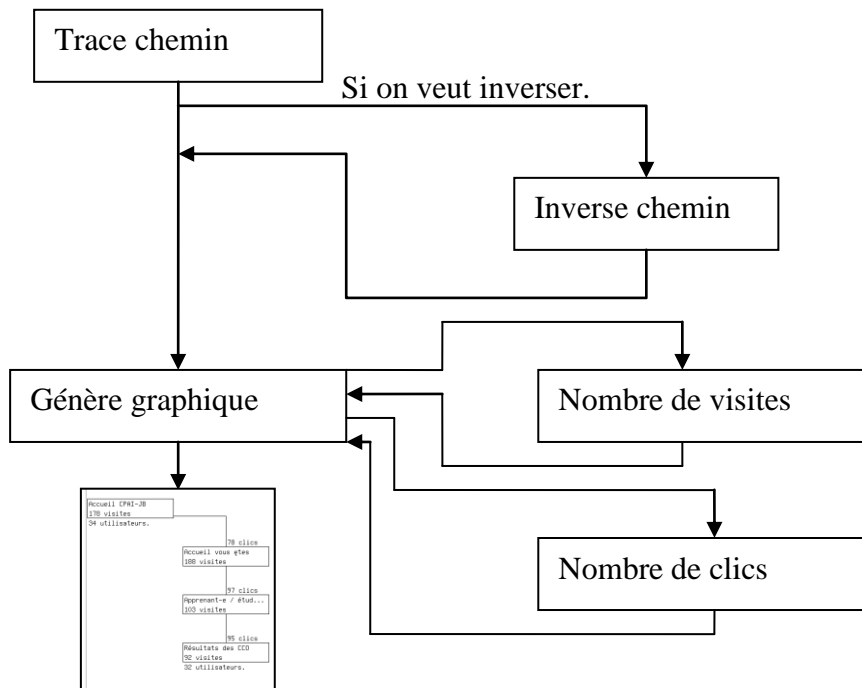


- index.php :** Page d'accueil.
Inclue :
 - config.inc.php
 - connect.inc.php
 - function.inc.php
 - option.php
 - graphique.php
 - gen_data_simple.php
- option.php :** fichier qui contient l'animation flash qui nous permet de modifier les paramètres.
- graphique.php :** fichier qui génère le graphique.
- gen_data_simple.php :** fichier qui nous génère les données simplifiées pour la base t_resume_stat.
- config.inc.php :** contient tous les paramètres de l'application.
- img :** dossier contenant la dernière image générée.
- connect.inc.php :** fichier de configuration de la connexion Mysql.
- function.inc.php :** fichier où se trouve toutes les fonctions qui nous sont utiles.
- option.swf :** application qui nous permet de modifier les options
- option fla :** code source de l'application pour les options.

2.2. Structure de l'application :



2.3. Déroulement de la procédure de génération du graphique :



3. Détail du travail.

3.1. Etude des technologies

Pour l'affichage, le fait de générer des images en PHP surcharge le serveur. La possibilité de générer l'aperçu avec un langage client comme flash (action Script) résoudrait le problème ; cependant, un autre problème se poserait, avec flash on ne peut pas sauvegarder les images. Vu la situation et surtout pour pouvoir sauvegarder les images (ex : pour l'historique) je vais générer la page (l'image) en php.

3.2. La librairie GD.

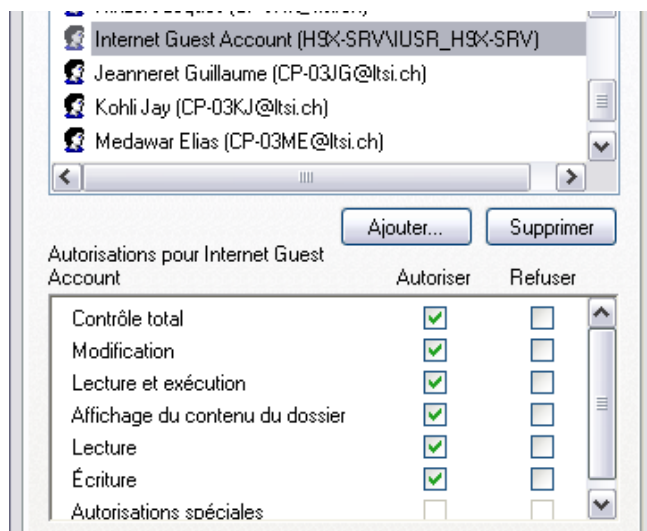
Cette librairie nous permet de générer des images en php. Cette librairie est livrée avec PHP, mais *elle n'est pas activée* car cette dernière demande beaucoup de ressource système et peut passablement ralentir un serveur.

Installation :

Pour pouvoir utiliser cette librairie, il faut l'activer : pour cela il faut enlever la virgule devant extension-php_gd2.dll dans le fichier php.ini. Ce fichier se trouve généralement dans le dossier Apache sur le serveur.

Configuration du dossier :

Pour pouvoir enregistrer les images dans un dossier, il faut donner les droits suivants à l'Internet Guest Account :



3.3. Nombre de visites.

Sur le graphique, il est demandé d'avoir le nombre de visites par page. Comme cette opération se répétera plusieurs fois, la fonction nb_visite a vu le jour. Cette fonction reçoit l'id de la page et nous renvoie le nombre de visites pour cette page.

Analyse :

Si la page étudiée =10

Résultat de la base de données.

User	Heure :	10 :22	10 :23	10 :24	10 :25	10 :26	10 :45	10 :46	10 :47
1 :	Page :	10	30	32	10	30	10	30	32
User	Heure :	10 :23	10 :23	10 :24	10 :25	10 :27	10 :28	10 :29	
2 :	Page :	10	30	10	30	10	30	32	

- =1 visite : l'intervalle de temps entre les deux visites est plus petite que 20 minutes.
- =1 visite : la dernière visite remonte à plus de 20 minutes.
- =1 visite : l'intervalle de temps entre les visites est plus petite que 20 minutes.

Total 3 visites.

3.4. Nombre de clics.

Afin de trouver le nombre de clic entre deux pages la fonction nb_click a été créée. Cette fonction reçoit l'id de la page de départ et l'id de la page d'arrivée. Un clic est pris en compte même si il se reproduit dans les 20 minutes qui suivent. Cette décision a été prise et discuté avec monsieur Hirtzel.

Explication du fonctionnement :

On sélectionne toutes les entrées dans la base.

On enregistre dans un tableau à 2 dimensions les données. La première dimension contient l'adresse IP du user. La deuxième, les références de la page. Ainsi nous avons pour chaque visiteur, les pages visitées dans l'ordre.

Pour chaque visiteur, on parcourt les visites. Si la page parcourue = la page de destination, et que la page parcourue avant = la page de départ, on incrémente le nombre de clic.

3.5. Chemins emprunté :

Afin de trouver les chemins empruntés par les visiteurs ainsi que combien de personnes ont emprunté les mêmes chemins, la fonction traceChemin a été créée. Cette fonction ne reçoit rien.

Explication du fonctionnement :

On sélectionne toutes les entrées dans la base
On trie par rapport à la date d'enregistrement.

On enregistre dans un tableau à 3 dimensions les données. La première dimension contient l'adresse IP du user. La deuxième, regroupe les heures de visites et les id de pages. La troisième, contient les heures et les ids des pages. Ainsi nous avons pour chaque visiteur les heures de visites pour chaque page.

Pour chaque visiteur, on parcourt les visites. Si l'intervalle de temps entre cette visite et celle d'avant est plus grand que 20 minutes, on dit que c'est une nouvelle trace. On regroupe les traces par leurs heures de visites de la page de départ.

On enlève les doublons parmi les chemins parcourus par un utilisateur (grâce à array unique). On colle le résultat dans un tableau où il y a le résultat de tous les visiteurs. Et ensuite, on enlève à nouveau les doublons parmi les chemins parcourus par les visiteurs. On obtient ainsi les différents chemins parcourus par l'ensemble des visiteurs.

On parcourt les différents chemins parcourus par tous les visiteurs. Pour chaque chemin, on regarde combien de fois on est passé sur ce chemin. On ne compte que les fois où l'intervalle de temps entre les passages sur ce chemin est plus grand que 20 minutes.

Problème rencontré :

Pour réaliser cette fonction il faut beaucoup travailler avec les tableaux à plusieurs dimensions et il est difficile de les manipuler sans bien comprendre leur contenu.

La solution a été de représenter sur papier les tableaux avec des données simplifiées.

Extrait des tableaux :

Si la page de départ = 10

Les différents chemins parcourus par les utilisateurs.

Résultat de la base de données.

User 1 :	Heure :	10 :22	10 :23	10 :24	10 :45	10 :46	13 :45	13 :46	13 :47
	Page :	10	30	32	10	30	10	30	32
User 2 :	Heure :	10 :23	10 :23	11 :24	11 :25	12 :27	12 :28	12 :29	
	Page :	10	30	10	30	10	30	32	

 =chemin 1: l'ordre de visite des pages est identique.


Ces pages ne font qu'un chemin même si ce n'est pas le même utilisateur qui fait ce chemin.


 =chemin 2

Nombre de fois qu'on emprunté un chemin.

Résultat de la base de données.

User 1 :	Heure :	10 :22	10 :23	10 :24	10 :45	10 :46	13 :45	13 :46	13 :47
	Page :	10	30	32	10	30	10	30	32
User 2 :	Heure :	10 :23	10 :23	11 :24	11 :25	12 :27	12 :28	12 :29	
	Page :	10	30	10	30	10	30	32	

 On a emprunté ce chemin 3 fois

 On a emprunté ce chemin 2 fois



Le user 1 l'a emprunté deux fois mais, l'intervalle de temps entre les visites, était de plus que 20 minutes. Le user 2 l'a emprunté deux fois mais, l'intervalle de temps entre les visites était de moins que 20 minutes. Donc on ne compte qu'une fois.

3.6. Génération du graphique :

Analyse :

1. On pourrait faire des modèles de graphique à la main et les remplir avec le contenu désiré.

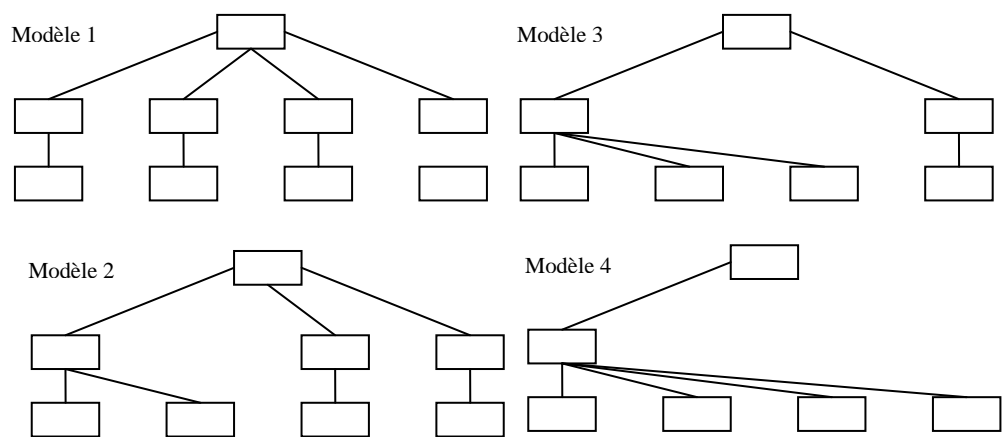
Avantage :

- On ne surcharge pas le serveur avec la création des images.

Défaut :

- Le nombre d'étage serait limité.
- Le nombre de chemins que l'on veut afficher serait limité.
- Le nombre de modèle à créer serait énorme.

Exemple de modèle :



2. On pourrait avoir des modèles d'éléments et tout dépendrait du cas on prendra un modèle différent. Les modèles seraient générés par un script PHP. Ainsi si un modèle nous manque, on le génère.

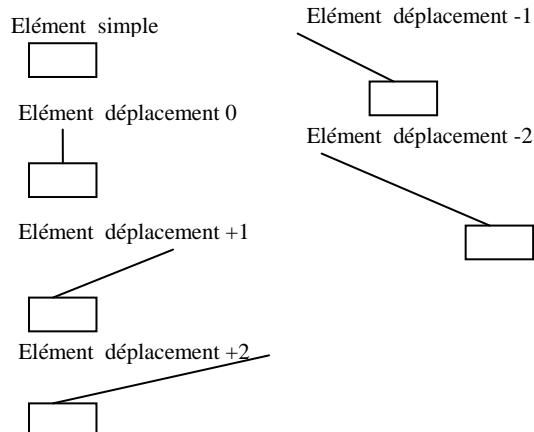
Avantage :

- On ne surcharge pas le serveur avec la création des images à chaque aperçu des statistiques

Défaut :

- On ne peut pas changer des options.

Exemple de modèle :



3. On peut tout générer en programmation.

Avantage :

- Vite fait.
- Un plus grand choix de paramètre.

Défaut :

- Surcharge le serveur.

La solution 3, est l'idéal pour notre cas. Elle est complètement modulable et si l'on veut changer des paramètres, on le peut facilement. Comme le nombre de personnes qui regarderont les statistiques est minime, le serveur ne sera pas trop surchargé.

Problème rencontré 1 :

Quand on a beaucoup de chemin, le temps de réponse du serveur est trop long. Ce qui provoque un « **time out** » et une erreur.



Solution :

1. On ajoute une variable dans le fichier de config, qui nous dit à partir de combien d'utilisateurs on affiche un chemin. (comme cela on peut masquer les chemins pris par 1 voir 2 visiteurs). Mais, on faussera un peu la statistique même si ces chemins ne nous indique pas grandes choses.
2. On affiche le graphique sur plusieurs pages ainsi, le temps de calcul est moins important.

Problème rencontré 2 :

Notre image ayant le même nom, le navigateur la met dans son cache et après on ne va plus chercher l'image que l'on régénère sur le serveur.

Solution :

Nommer l'image d'après la date, heure, jour, mois, minutes, secondes. Et à chaque visite, on affiche une image avec un autre nom, et on efface toutes les images dans le dossier ./img comme cela on ne gaspille pas de la place pour rien.

Explication du fonctionnement :

On parcourt tous les chemins qu'on a reçus. Si on atteint le nombre maximum de chemins par étage, on ajout à la position _y la hauteur de cet étage.
On parcourt les points de chaque chemin ; si on n'a pas encore affiché ce point à cette hauteur et dans le chemin d'avant. De même que si la partie de chemin jusqu'à ce point n'est pas identique à la même partie du chemin d'avant jusqu'à cette hauteur, on affiche le point. Dans le cas contraire on affiche un lien au point qui est à cette hauteur.
Si l'on a atteint le dernier point de ce chemin, on indique le nombre d'utilisateurs qui ont pris ce chemin.
On enregistre l'image.

3.7. Génération du cube OLAP :

Analyse:

Généralité¹:

Un **cube OLAP** est une représentation abstraite d'informations multidimensionnelles exclusivement numériques utilisées par l'approche OLAP (acronyme de *On-line Analytical Processing*). Cette structure est prévue à des fins d'analyses interactives par une ou plusieurs personnes (souvent ni informaticiens ni statisticiens) du métier que ces données sont censées représenter.

Les 12 règles d'un cube OLAP sont :

- Vue conceptuelle multidimensionnelle
- Transparence
- Accessibilité
- Constance des temps de réponses
- Architecture Client/serveur
- Indépendance des dimensions
- Gestion des matrices creuses
- Accès multi-utilisateurs
- Pas de restrictions sur les opérations inter et intra dimensions
- Manipulation des données aisée
- Simplicité des rapports
- Nombre illimité de dimensions et nombre illimité d'éléments sur les dimensions

¹ Source www.wikipedia.com .

Création :

Pour créer un cube olap, il faut pouvoir se connecter sur un serveur qui contient les données dans notre cas, c'est www.cpaijb.ch . Comme le serveur est un serveur mysql, il va falloir faire un mysql connecteur ODBC. Le connecteur est trouvable sur le net. (<http://dev.mysql.com/downloads/connector/odbc/3.51.html>)

Ainsi, grâce à MSQRY32.exe nous pouvons générer notre cube.

Utilisation :

Le cube peut être utilisé comme source de données dans Excel .D'où la possibilité de faire toutes les manipulations qu'offre Excel (graphique, etc...).

Préparation des données :

Comme on ne pourra pas avec des commandes mysql retrouver les visites (trop de paramètres à prendre en compte), on va créer une base de données simplifiées qui nous prépare les donnés à passer au cube. Cette table est générée par un script PHP que l'on doit lancer pour mettre à jour assez fréquemment².

Schéma de la base.

	Champ	Type	Attributs	Null	Défaut	Extra	Action					
<input type="checkbox"/>	id_stat	int(11)		Non		auto_increment						
<input type="checkbox"/>	nom_page	varchar(200)		Non								
<input type="checkbox"/>	resolution_user	varchar(15)		Non								
<input type="checkbox"/>	adresse_acces	varchar(120)		Non								
<input type="checkbox"/>	os_user	varchar(30)		Non								
<input type="checkbox"/>	navigateur_user	varchar(30)		Non								
<input type="checkbox"/>	domaine_user	varchar(100)		Non								
<input type="checkbox"/>	clic	tinyint(4)		Non	1							
<input type="checkbox"/>	Anne	year(4)		Non	0000							
<input type="checkbox"/>	Mois	varchar(20)		Non								
<input type="checkbox"/>	Jours	varchar(20)		Non								
<input type="checkbox"/>	Heure	time		Non	00:00:00							

Afin d'avoir des noms de domaines qui signifient quelque chose, la fonction findDomaine à été fait. Elle ressort seulement le domaine, sans l'adresse IP.

Exemple :

adsl-84-227-54-45.adslplus.ch devient **adslplus.ch**.

adsl-102-17-23-25.adslplus.ch devient **adslplus.ch**.

² Possibilité d'automatiser cette étape mais pas demander dans le cadre du travail.

Création du Driver :

Après l'installation du connecteur,

Dans panneau de configuration>Outils d'administration>source de donn  ODBC, on ajoute un driver mysql.

Avec les param tres suivants :

The screenshot shows the 'Connect Options' tab of the ODBC Data Source Administrator. The fields are filled with the following values:

Field	Value
Data Source Name	www.cpaijb.ch
Description	Base de donn� CPAIJB
Server	www.cpaijb.ch
User	remote-admin
Password	*****
Database	cpaijb

The 'Optional' checkbox is checked, and the 'Default' checkbox is unchecked. The 'Description' field contains the text 'Base de donn  CPAIJB'. The 'Test' button is highlighted.

Password = *****.

Probl me 1 :

Le port qu'on utilise (3306) est bloqu  par le proxy de l' cole.

Solution :

Suite   une demande M.Musy a d bloqu  ce port pour acc der au site de l' cole.

Probl me 2 :

L'utilisateur qui veut acc der aux donn es, doit pouvoir faire les requ tes depuis une machine distante. Donc il a fallu trouver le probl me et apr s cr er un utilisateur, qui peut ex cuter les commandes, depuis une machine distante.

Création du cube :

Après la création du connecteur, on importe les données dans MSQRY32.EXE grâce à des commandes mysql. Ici, on a profité pour donner des noms significatifs aux champs.

Requête mysql :

```
SELECT t_resume_stat_0.id_stat, t_resume_stat_0.nom_page AS 'Nom de la
page', t_resume_stat_0.resolution_user AS 'Résolution',
t_resume_stat_0.adresse_acces AS 'Adresse précédent cette visite',
t_resume_stat_0.os_user AS 'Système d'exploitation',
t_resume_stat_0.navigateur_user AS 'Navigateur ',
t_resume_stat_0.domaine_user AS 'Domaine', t_resume_stat_0.clic,
t_resume_stat_0.Anne AS 'Année', t_resume_stat_0.Mois AS 'Mois',
t_resume_stat_0.Jours, t_resume_stat_0.Heure AS 'Heures'
FROM cpaijb.t_resume_stat t_resume_stat_0 ;
```

Sauvegarde des paramètres et des données:

stat_cpaijb.qry : contient la requête mysql. (S'il faut refaire le cube.)


stat_cpaijb.dqy : Les paramètres de Microsoft Query. (S'il faut refaire le cube.)

stat_cpaijb.oqy : C'est le fichier de requête pour Excel. Il contient un lien jusqu'au fichier cube donc si on change d'emplacement du fichier cube, Excel nous demandera de redire où se trouve le fichier cube. Ce fichier peut être ouvert et affiché par plusieurs personnes en même temps. Mais on ne peut pas depuis Excel actualiser les données pendant qu'une autre personne utilise le cube.

stat_cpaijb.cub : Ce fichier contient toutes les données. Ce dernier, peut être mis à jour quand une seule personne depuis Excel met à jour les informations.

Utilisation sous Excel :

Pour intégrer le cube dans Excel, on clique sur stat_cpaijb.oqy ou sous Excel on fait Données>Données Externes>Importer et on choisit notre stat_cpaijb.oqy .

Pour mettre à jour les données du cube, il faut presser sur Actualiser les données. 

Il est important que le connecteur ODBC soit installé chez l'utilisateur.

Ensuite libre à chaque personne d'afficher les informations comme bon lui semble.

Exemple d'interprétation possible :

Les interprétations sont innombrables, car tout dépend de quel axe on regarde, les informations peuvent signifiées toutes autres choses. Et dans notre cas, nous avons 11 axes que l'on peut manipuler à notre guise. Voici quelques interprétations que pourrait faire le webmaster.

Navigateur :

Somme De clic	
Navigateur	Total
msie_6	76.24%
firefox	21.54%
safari	1.53%
msie_5.5	0.32%
mozilla_5	0.13%
msie_5	0.13%
mozilla_4	0.11%
Total	100.00%

Cet exemple nous indique quels sont les navigateurs les plus utilisés sur notre site. Ces informations sont très importantes, car la compatibilité entre les navigateurs et très mince. Donc si on a beaucoup de gens, qui viennent avec Firefox, on devrait faire 2 versions de notre site (ou de leur feuille de style) pour ainsi satisfaire tout nos visiteurs.

Résolution :

Somme De clic	
Résolution	Total
1024 x 768	46.66%
1280 x 1024	37.05%
1152 x 864	4.08%
800 x 600	3.83%
1280 x 800	2.46%
1600 x 1200	1.59%
1280 x 960	1.20%
1400 x 1050	1.15%
1680 x 1050	1.01%
1280 x 768	0.96%
Total	100.00%

Cet exemple nous indique les résolutions. Nous pouvons ainsi avoir un aperçu du site de la même manière que la plus part de nos visiteurs. Notre site a un problème quand nous avons une résolution de 800 x 600, mais comme cela représente que 3.83% des visiteurs nous pouvons être tranquilles.

Heure de visite :

Somme De clic	
Heure	Total
2	1
4	1
5	2
1	17
6	36
0	48
7	54
23	74
8	94
9	146
14	159
10	163
12	165
Total	1144

Cet exemple nous indique les heures les moins visitées ainsi nous pouvons définir des heures de back up où des heures pour faire de la maintenance.

4. Conclusion :

Le cahier des charges a été tenu et réalisé entièrement. Le code est fait de manière à ce qu'on puisse ajouter d'autres fonctionnalités sans devoir tout refaire. Une interface pour gérer les options du graphique a été faite en plus. Cette interface a pour but de pouvoir mieux cibler les heures, dates qu'on l'on a envi d'étudier. De pouvoir afficher les dernières pages qu'un visiteur a visité et de savoir par où il est passé.

J'ai beaucoup appris en faisant ce travail non seulement en programmation mais aussi en ergonomie du Web et en communication avec d'autre plateforme que le PHP simple dans son coin.

Le résultat est enrichissant pour un programmeur Web comme moi. Non seulement on voit ce qui se passe sur notre site, mais aussi on voit des informations utiles pour d'autres sites : tel que l'importance d'une bonne structure, les standards à utiliser, ce que veulent les gens sur un site.

J'ai fais beaucoup de site dans le cadre de mon apprentissage sans bien avoir l'avis d'un visiteur normal qui ne s'y connaît pas vraiment en informatique. A travers cette application, j'ai l'impression d'avoir un contact avec le visiteur et de pouvoir répondre à ses attentes.