



Rapport de stage

STAGE AU CC-IN2P3

Centre de Calcul de l'Institut National de Physique Nucléaire et de Physique des Particules

Du 23 mai au 1^{er} juillet 2022

CC-IN2P3, 21 Av. Pierre de Coubertin, 69100 Villeurbanne

<https://cc.IN2P3.fr/>

Oscar Jundt-Schmitter | BTS SNIR
Lycée Edouard Branly, Lyon 5e

Table des matières

<i>Mes remerciements</i>	2
<i>1.Présentation de l'entreprise</i>	2
1.1.presentation générale	2
1.2.situation géographique	3
1.3.structure juridique	4
1.4.finances	4
<i>2.Présentation du service d'accueil du stage</i>	5
2.1.Organisation	5
2.2.Activités du service.....	5
<i>3.Activité durant le stage</i>	5
3.1. Présentation générale.....	5
3.2.Présentation détaillée d'une page web (une page detaillee)	8
<i>4.Réflexions et conclusion</i>	15
<i>5.Annexe : Certificat de stage</i>	16
<i>6.Annexe : Document Anglais</i>	16
<i>7.Annexe : autre</i>	17
7.1.photo salle serveur	17
7.1.1.photo salle serveur 1.....	17
7.1.3.Photo Salle serveur 2.....	18
7.2.photo carte.....	19
7.3.les organigrammes	20
7.3.1.organigramme (pour CC-IN2P3) général.....	20
7.3.2.organigramme (pour CC-IN2P3) précis.....	21
7.4.liens utilises pour le projet	22

Mes remerciements

Tout d'abord je voudrais remercier le CC-IN2P3, et notamment Monsieur Kachelhoffer, ingénieur-chercheur, mon maître de stage, pour m'avoir pris sous son aile et aussi pour m'avoir encadré. Je voudrais remercier tout le Service Support pour m'avoir accueilli.

Je voudrais remercier le Lycée Edouard Branly et ses professeurs pour m'avoir encadré et m'avoir fourni la convention de stage et notamment mon professeur référent Monsieur Brugère, qui s'est déplacé pour le rendez-vous avec mon maître de stage au CC-IN2P3. Je voudrais aussi remercier le jury qui m'accorde une partie de son temps pour lire et évaluer mon rapport.

1. Présentation de l'entreprise

1.1. PRESENTATION GENERALE

Présentation du cc-in2p3 :

Le CC-IN2P3 est le Centre de calcul de l'IN2P3 (Institut national de physique nucléaire et de physique des particules). Le CC-IN2P3 est un service informatique **fondé en 1986** et dirigé par **Pierre-Etienne Macchi**. Son siège social est à **Villeurbanne**. Il fournit **des traitements de données, calculs scientifiques, Cloud, stockage, hébergement et réseaux** à très haut débit. Le CC-IN2P3 a **4000 utilisateurs répartis en 70 groupes** de recherche, qui utilisent ses services pour mener à bien leurs expériences de science physique. Dans les projets connus, il est possible de citer par exemple : le CERN ou les accélérateur de particule.

Le CC-IN2P3 est une administration qui compte **86 agents**, qui ont des statuts variés : **1 chercheur permanent, 63 IT, 19 CDD, 1 CDI et 1 apprenti**. Le CC-IN2P3 a des salles de stockage (data center) dans **2 bâtiments** et un bâtiment supplémentaire est en cours de projet. En tout, le CC-IN2P3 stocke **340 Pétaoctets de données**.

Présentation de l'in2p3 :

Le Centre de calcul appartient à l'IN2P3 (Institut nationale de physique nucléaire et de physique des particules). L'IN2P3 a été **créé en 1971**, il a un effectif de **1000 chercheurs, 700 chercheurs sous contrat et 1500 autres personnels permanents**. L'IN2P3 se compose de **25 laboratoires et 15 laboratoires internationaux**. Les Instituts de l'IN2P3 sont répartis dans tout le pays.

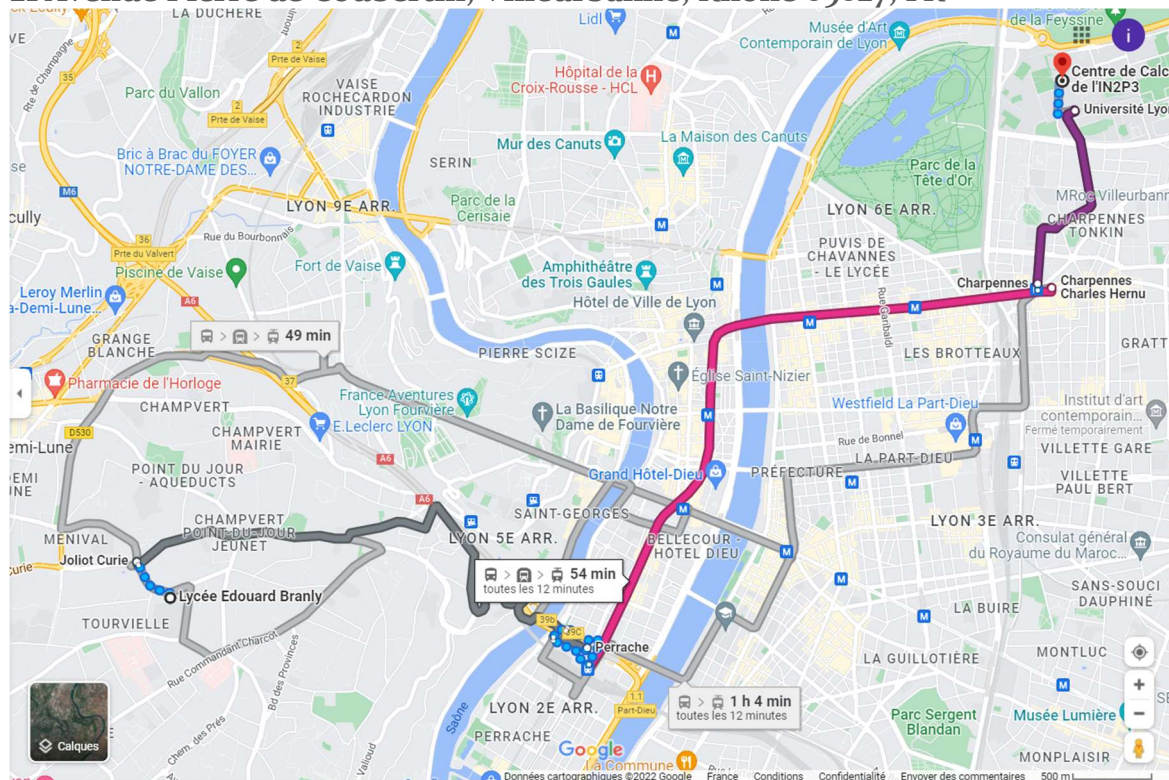
Présentation du CNRS

L'IN2P3 appartient lui-même au CNRS (Centre nationale de recherche scientifique). Le CNRS possède 10 instituts, dont l'IN2P3. Il avait en 2020 l'effectif suivant : 3200 personnels, 24000 permanents, dont 11000 chercheurs et 13000 ingénieurs.

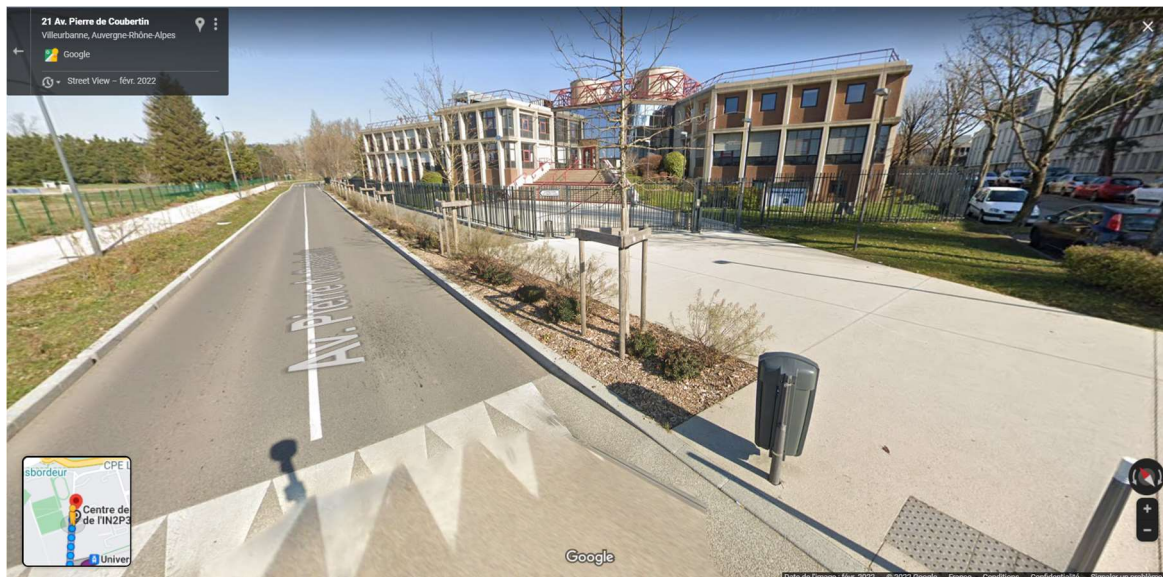
1.2.SITUATION GEOGRAPHIQUE

Comme indiqué ci-dessous, le Centre de calcul de l'IN2P3 est situé à l'est du Parc de la Tête d'Or, dans le complexe universitaire de la Doua.

21 Avenue Pierre de Coubertin, Villeurbanne, Rhône 69627, FR



Voici le complexe de l'IN2P3, un bâtiment englobant des bureaux et les salles pour les serveurs de calcul.



Concernant les transports qui desservent le CC-IN2P3, voir l'annexe 7.2.

1.3. STRUCTURE JURIDIQUE

Concernant l'organigramme du Centre de calcul de l'IN2P3, voir l'annexe 7.3.

Le Centre de calcul de l'IN2P3 appartient à l'IN2P3 (Institut national de physique nucléaire et physique des particules), qui appartient au Centre national de recherche scientifique (CNRS), qui dépend du Ministère de l'éducation et de la recherche.

Le Centre de calcul de l'IN2P3 est donc un organisme public, donc les employés du CC-IN2P3 sont des fonctionnaires ou des contractuels de droit public.

1.4. FINANCES

Pour tout ce qui concerne les finances, le CC-IN2P3 est dépendant de l'IN2P3, qui a un **budget annuel de 71M€**. Ce budget est réparti dans les différents centres de l'IN2P3. Le **budget en 2020 du CC-IN2P3** était de **8801K €**. Le budget de l'IN2P3 est attribué par le CNRS, sachant que le budget de CNRS en 2018 était de **3.4milliards d'euros**.

2. Présentation du service d'accueil du stage

2.1. ORGANISATION

Responsable du Service Support : David Bouvet.

2.2. ACTIVITES DU SERVICE

Le service dans lequel j'ai fait mon stage était le **Service Support** qui notamment s'occupe **d'aider et de faciliter l'utilisation du Centre de calcul**. Ce service peut notamment créer des plateformes de monitoring pour faciliter l'utilisation du data center. C'est dans ce cadre que j'ai exercé ma mission de stage.

3. Activité durant le stage

3.1. PRESENTATION GENERALE

Mon objectif de stage a été de procéder à la refonte d'un site de monitoring.

Présentation du site, objectif et contrainte :

Ce site de monitoring est constitué de tableaux, avec des données récupérées dans certaines bases de données du CC-IN2P3, et aussi de graphes. J'ai fait le développement des tableaux mais pas des graphes. Ces tableaux contiennent des données à propos des data center et des groupes de recherche des utilisateurs. Mon objectif est de procéder à la refonte du site. Cela veut dire remettre les anciennes pages à neuf, faire de nouvelles pages, remettre la structure du site à neuf, réorganiser le traitement des données et faire de nouvelles mises en forme dans des tableaux (en supprimant les anciennes mises en forme). Mon objectif était aussi de trouver une nouvelle méthode pour le traitement des données et la mise en forme des données, et de faciliter la maintenance. Il me fallait aussi remplacer pconnect par pdo.

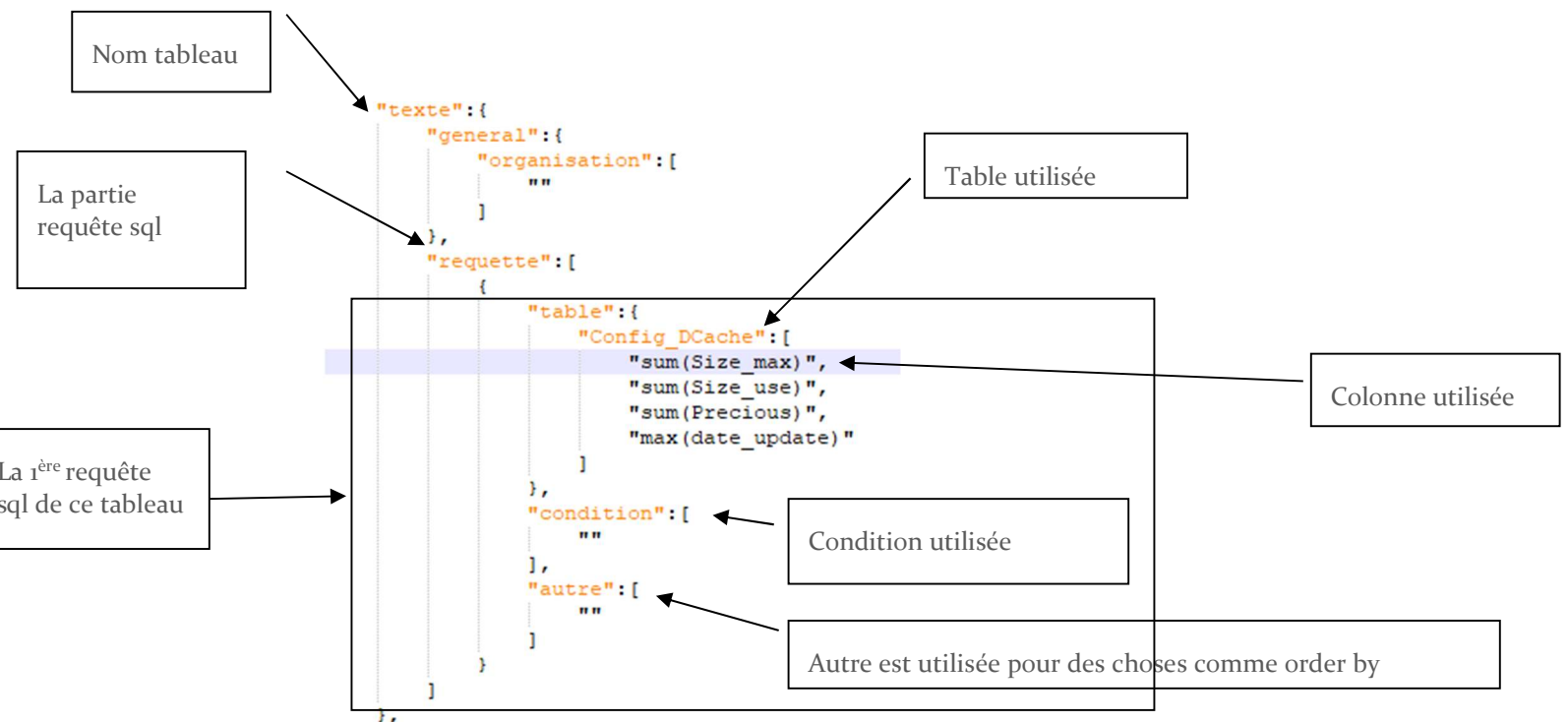
Les langage et logiciel utiliser pour mon travail :

Pour effectuer ces missions, j'ai utilisé comme langages **le html** pour la structure du site, **le css** pour le style du site, **js** pour les interactions dans le site, **le php** pour le développement côté serveur, **mysql** pour le traitement des données et **json** en tant que fichier de paramétrage. J'ai utilisé json notamment pour paramétrer les requêtes sql, les tableaux, les données et récupérer des requêtes sql. J'ai aussi utilisé un ide (Integrated Development Environment) spécifique qui est visual studio code. J'ai utilisé cet ide car c'est celui j'utilise habituellement pour du développement web. Pour tout ce qui est visualisation des base de données, j'ai utilisé **phpmyadmin**, qui est le gestionnaire de base de données de php.

Présentation du json et de son utilité :

Pour faciliter la maintenance, j'ai créé des fichiers de configuration json (un fichier par page web). J'ai aussi créé des scripts php, comme par exemple un script pour les fonctions (ce script sert pour convoquer toutes les fonctions que j'ai créées), un script pour les class, etc.

Pour certaines des actions (ex. générer une requête sql), on utilise un fichier de paramétrage qui est écrit en json. Ce fichier sert principalement pour les requêtes sql, la mise en forme des données et le header du tableau (c'est-à-dire les titres des colonnes du tableau). Pour cela il m'a fallu respecter une structure, indiquée ci-dessous.



Pour trouver la structure correcte (c'est-à-dire la plus optimale), j'ai pris du temps avant de trouver (voir la structure ci-dessus). La partie **organisation** est la partie qui s'occupe de réorganiser les données reçues de requêtes sql, notamment en précisant si c'est du float (nombre à virgule), int (nombre entier) ou autre (string). Le script json précise aussi combien il y a de nombres après la virgule pour les float. Il donne aussi le titre des colonnes, puis, pour la partie **requete**, il donne toutes les données pour faire la requête :

le nom de la table (la Config_DCache), les valeurs à prendre dans la table, par exemple `sum(Size_max)`, c'est-à-dire la somme de Size_max, puis les conditions (le where dans la requête). Il faut préciser que pour un tableau il peut y avoir plusieurs requêtes sql dans la partie requette, comme dans cet exemple ci-dessous.

```

//
"requete": [
  {
    "table": {
      "Info_Storage": [
        "Sum(space_used/1024/1024/1024)",
        "insert_time"
      ],
      "condition": [
        "type='pbs' AND site='ccin2p3' AND storage_role='throng'"
      ],
      "autre": [
        "GROUP BY storage_role"
      ]
    },
    {
      "table": {
        "Info_Quota": [
          "Sum(group_allocated_space/1024/1024/1024)"
        ],
        "condition": [
          "type='pbs' AND site='ccin2p3'"
        ],
        "autre": [
          ""
        ]
      },
      {
        "table": {
          "Info_Storage": [
            "unix_group",
            "Sum(space_used/1024/1024/1024)",
            "allocated_space_to_group/1024/1024/1024"
          ],
          "condition": [
            "type='pbs' AND site='ccin2p3' AND storage_role='throng'"
          ],
          "autre": [
            "GROUP BY unix_group ORDER BY unix_group ASC"
          ]
        }
      ]
    }
  ]
}

```

Position 0 : requette 1

Position 1 : requette 2

On voit donc qu'il y a pour un tableau plusieurs requêtes sql. Les données pour constituer la requête sql ont toutes la même syntaxe dans json (voir ci-dessus), mais elles sont séparées par des accolades {}. Chaque requête est encadrée par des {} pour différencier les différentes requête sql et connaître leur position dans le json.

Présentation des fonction et classe utiliser et créer :

Pour faire la refonte du site, j'ai dû créer mes propres fonctions et class en php. Notamment deux class pour la construction des tableaux de données, l'une servait pour la création et la manipulation des lignes (*class ligne*) du tableaux et l'autre pour la création

et la manipulation du tableaux (*class tableaux*). Les deux sont reliées par des fonctions, notamment la fonction créerligne de la class tableaux. Puis j'ai dû faire des fonctions pour le traitement des données et la génération des requêtes sql.

Pour la génération des requêtes sql, il existe deux fonctions principales :

- ***donnee_requette(\$name_page,\$name_tableaux,\$numero)***, cette fonction sert à récupérer les données d'une requête sql ; à la fin, elle retourne une liste avec en position 0, les colonnes de la table à récupérer, en position 1, le nom de la table, en position 2, les conditions ;
- ***envoi_requette(\$liste_pdo,\$name_page)***, cette fonction sert grâce à la fonction *donnee_requette* à former la structure des requêtes de type select (donc select <nom colonne> from <nom table> where <condition>) de la page puis on envoie grâce au pdo, la requête à la base de données et on récupère les données et on les met dans une liste, ce sont les résultats. Cette liste est de type dictionnaire, avec une clef qui est le nom du tableau et une valeur qui est le résultat de la requête sql. Pour les autres fonctions, ce sont des fonctions pour le traitement des données et la mise en forme, comme par exemple la fonction ***signalement_critique()*** qui inspecte si la valeur est critique ou pas, pour notamment la mettre en rouge dans le tableau, ou la fonction ***organisation_donnee(\$nbcolone,\$id_tableaux)*** qui notamment réorganise les données, via le json.

Remplacement de pconnect par pdo :

pconnect est à remplacer par pdo, car pdo est plus récent que pconnect, qui est maintenant obsolète.

Pour le remplacement de pconnect par pdo, j'ai dû aller voir toutes les pages web du site pour remplacer directement pconnect par pdo. Puis, pour la connexion à la data base, j'ai créé un script qui est convoqué par les pages qui en ont besoin. Le pdo fonctionne comme une class : « ***\$pdo = new PDO(\$SQL_DSN,\$SQL_LOGIN,\$SQL_PASS);*** » cette partie de code sert à faire la connexion à la data base.

\$SQL_DSN : le nom de la base de données

,\$SQL_LOGIN : l'identifiant

,\$SQL_PASS : le password

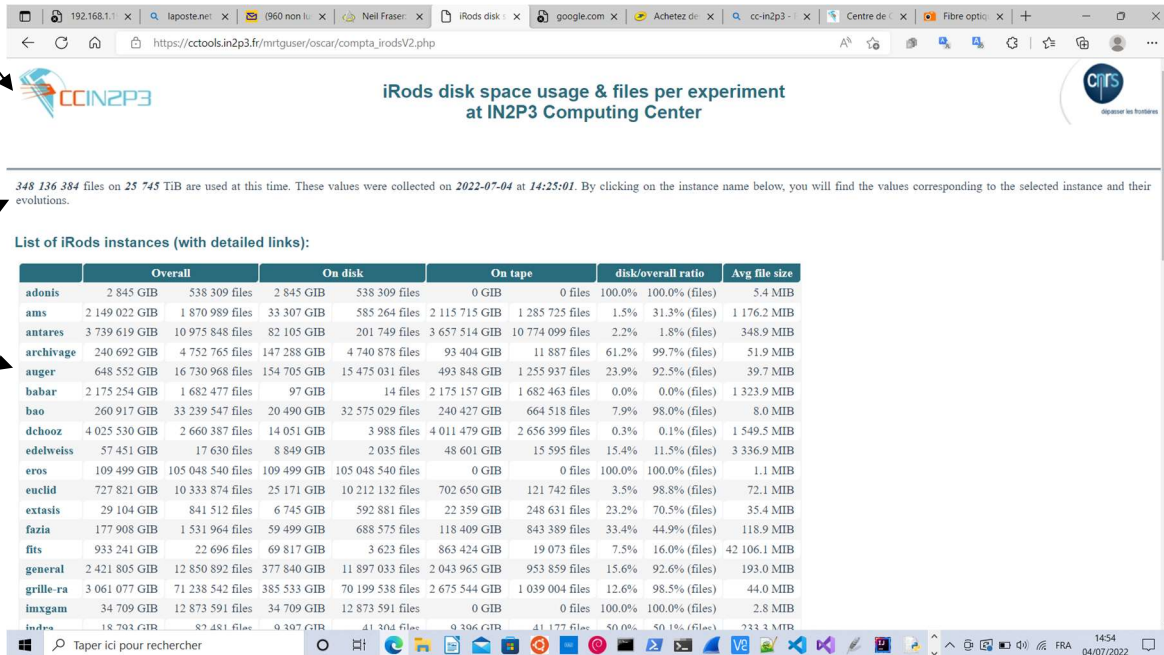
3.2.PRESENTATION DETAILLEE D'UNE PAGE WEB (UNE PAGE DETAILLEE)

Voici ci-dessous la page web utilisée comme exemple :

Le header

Le texte d'intro

Le tableau



	Overall	On disk	On tape	disk/overall ratio	Avg file size
adonis	2 845 GIB 538 309 files	2 845 GIB 538 309 files	0 GIB 0 files	100.0% 100.0% (files)	5.4 MiB
ams	2 149 022 GIB 1 870 989 files	33 307 GIB 585 264 files	2 115 715 GIB 1 285 725 files	1.5% 31.3% (files)	1 176.2 MiB
antares	3 739 619 GIB 10 975 848 files	82 105 GIB 201 749 files	3 657 514 GIB 10 774 099 files	2.2% 1.8% (files)	348.9 MiB
archivage	240 692 GIB 4 752 765 files	147 288 GIB 4 740 878 files	93 404 GIB 11 887 files	61.2% 99.7% (files)	51.9 MiB
auger	648 552 GIB 16 730 968 files	154 705 GIB 15 475 031 files	493 848 GIB 1 255 937 files	23.9% 92.5% (files)	39.7 MiB
babar	2 175 254 GIB 1 682 477 files	97 GIB 14 files	2 175 157 GIB 1 682 463 files	0.0% 0.0% (files)	1 323.9 MiB
bao	260 917 GIB 33 239 547 files	20 490 GIB 32 575 029 files	240 427 GIB 664 518 files	7.9% 98.0% (files)	8.0 MiB
dchooz	4 025 530 GIB 2 660 387 files	14 051 GIB 3 988 files	4 011 479 GIB 2 656 399 files	0.3% 0.1% (files)	1 549.5 MiB
edelweiss	57 451 GIB 17 630 files	8 849 GIB 2 035 files	48 601 GIB 15 595 files	15.4% 11.5% (files)	3 336.9 MiB
eros	109 499 GIB 105 048 540 files	109 499 GIB 105 048 540 files	0 GIB 0 files	100.0% 100.0% (files)	1.1 MiB
euclid	727 821 GIB 10 333 874 files	25 171 GIB 10 212 132 files	702 650 GIB 121 742 files	3.5% 98.8% (files)	72.1 MiB
extasis	29 104 GIB 841 512 files	6 745 GIB 592 881 files	22 359 GIB 248 631 files	23.2% 70.5% (files)	35.4 MiB
fazia	177 908 GIB 1 531 964 files	59 499 GIB 688 575 files	118 409 GIB 843 389 files	33.4% 44.9% (files)	118.9 MiB
fits	933 241 GIB 22 696 files	69 817 GIB 3 623 files	863 424 GIB 19 073 files	7.5% 16.0% (files)	42 106.1 MiB
general	2 421 805 GIB 12 850 892 files	377 840 GIB 11 897 033 files	2 043 965 GIB 953 859 files	15.6% 92.6% (files)	193.0 MiB
grille-ra	3 061 077 GIB 71 238 542 files	385 533 GIB 70 199 538 files	2 675 544 GIB 1 039 004 files	12.6% 98.5% (files)	44.0 MiB
imxgam	34 709 GIB 12 873 591 files	34 709 GIB 12 873 591 files	0 GIB 0 files	100.0% 100.0% (files)	2.8 MiB
indus	18 703 GIB 83 881 files	0 107 GIB 41 304 files	0 306 GIB 41 177 files	50.0% 50.1% (files)	733.3 MiB

les différent partie du site, head, header, section

Pour la structure de la page, on aperçoit d'abord le head qui se compose de balises invisibles, qui n'apparaîtront pas dans la page web. Dans le fichier de développement de la page web, on inclut un script qui va intégrer le head dans la page web. J'ai séparé le script head de la page web parce que ce script est utilisé dans plusieurs autres pages, alors la maintenance se fait plus facilement. Dans la page web, il y a aussi une variable php appelée name qui sert à récupérer le nom de fichier de la page web. Dans l'exemple de cette page : « `compta_irodsV2` ». Cette méthode sert à savoir quelle json utiliser car le json de la page a le même nom que la page web.

```
<head>
<?php
//inclue la liste des titre du site
//definit le nom de la page
$name_page=basename(__FILE__, '.php');
//inclue le head
include_once "php/head.php";
?>
</head>
```

On inclue un script,
Include_once '<chemin/fichier>'

Voici ci-dessous le script `head.php` :

```

<!--recuperation du titre de la page en fonction du nom de la page qui est dans la liste dans le fichier liste_hl.php-->
<title>
<?php
//on recupere dans le json de la page, le titre de l'onglet de la page web
$file = 'json/'.$name_page.'.json';
$data = file_get_contents($file);
$objheader = json_decode($data);
echo $objheader->general->titre;
?>
</title>
<!--on met en utf-8 les page web-->
<meta charset="utf-8"/>
<!--<meta name="viewport" content="width=device-width, user-scalable=no"/>-->
<!--on specifie les chemin pour incorporer la partie js et css-->
<link rel="stylesheet" href="css/style.css" type="text/css"/>
<script src="js/script.js"></script>
<?php
if(strstr($name_page,"irods")==false || isset($_GET['instance'])) {
    print("<script language='JavaScript' src='../js/tablesort.js?id=<?php rand()??>'></script>");
}
?>

```

On récupère dans le json de la page, le titre de la page

Ensuite, dans la page web se trouve le bloc body, dans lequel on trouve tout ce qui apparaît dans la page web (sauf les éléments de langage php). Le body est divisé en deux blocs : le header (partie haute de la page web, statique) et la section (contenu de la page web ; la section est délimitée par une balise <div> dont l'identifiant est « milieu »). Pour la partie header, je procède de la même manière que pour la partie head. Cela signifie que j'inclue aussi un script dans la page web, qui s'appelle header.php. Dans ce script se trouve la structure du header. Vous trouverez ci-dessous la structure du header dans la page web.

```

<div id="haut">
<?php include_once "php/header.php";?>
</div>

```

L'identifiant du header

Voici ci-dessous le contenu de script **header.php**.

```

<!--page pour initialiser le header-->
<?php
print( "<div id='bandrole'\n" );
//on affiche les image du logo du cc-in2p3
print( "<a href='https://cc.in2p3.fr/'><img id='gauche' alt='logo of ccin2p3' src='img/ccv2.jpg'/></a>\n" );

//on recupere dans le page de la page le titre de la page web
$file = 'json/' . $name_page . '.json';
$data = file_get_contents( $file );
$objheader = json_decode( $data );

if ( $name_page == "info_manips_detailV2" ) {
    echo "<h1>" . $_REQUEST['group'] . " " . $objheader->general->titre . "</h1>";
} else {
    echo "<h1>" . $objheader->general->titre . "</h1>";
}

//on affiche l'image du logo du CNRS
print( "<a href='https://www.cnrs.fr/fr/page-daccueil'\><img id='droite' alt='logo of CNRS' src='img/cnrsV2.jpg'/></a>\n" );
print( "</div>\n" );
-?>

```

Méthode d'affichage de balise html en php

Méthode de récupération du fichier json

Pour récupérer le fichier json, par exemple dans les scripts ou dans les pages web, un élément est nécessaire : le chemin d'accès au fichier json. Pour cela il faut utiliser la fonction **file_get_contents(\$file)** pour récupérer le contenu du fichier et de **json_decode**, pour notamment récupérer le contenu json sous forme de liste php.

Tout le contenu de la page web, notamment les tableaux qui sont à l'intérieur, sont contenus dans la balise **<div>**, qui a comme identifiant « milieu ».

La première chose que j'ai fait dans la section, c'est d'inclure tous les scripts dont j'aurai besoin. Pour cela, dans cette page, j'appelle le scripte **include.php**, dans lequel se trouvent tous les scripts à inclure dans la page web. Voir ci-dessous.

```

<?php
/*
#####
fichier pour la liste des fichier a inclure
dans chacune des page web qui on des tableaux
#####
*/
$chemin="php/librairie";
//inclue les fonction utiliser
include_once "$chemin/fonction_php.php";
//inclue la class tableaux
include_once "$chemin/tableaux.php";
//inclue le code pour la connection a la BD
include_once "$chemin/connection_db.php";
//inclue les variable global
include_once "$chemin/variable_global.php";
//inclue la conversion d'un json en array
include_once "$chemin/config_json.php";
-?>

```

Voici ci-dessus le contenu du scripte **include.php**.

Puis, après avoir récupéré toutes les include, on va utiliser la fonction **envoi_requette(\$liste_pdo,\$name_page)** pour construire les requêtes sql, les envoyer et récupérer la réponse (données à mettre dans le tableau).

```

//
"tableaux1":{
  "general":{
    "organisation":[
      [" ","nocase","",""," ",1],
      ["Overall","nocase","",""," ",2],
      ["On disk","nocase","",""," ",2],
      ["On tape","nocase","",""," ",2],
      ["disk/overall ratio","nocase","",""," ",2],
      ["Avg file size","nocase","",""," ",1]
    ]
  },
  "requete":{
    {
      "table":{
        "Config_iRods"
      },
      "condition":{
        "type='serveurs'"
      },
      "autre":{
        "GROUP BY `Group` ORDER BY `Group` ASC"
      }
    }
  }
},

```

Partie
requête

1^{ère} requête

Table utilisée et colonne
utilisée * c'est tout

Les conditions (le
where) de la requête

Voici le json du premier tableau de la page prise en exemple. En voyant les données du json, on en déduit que la requête sql est :

« **select * from Config_iRods where type='serveurs' GROUP BY `Group` ORDER BY `Group` ASC** »

Donc la fonction va envoyer cette requête sql et le résultat sera récupéré dans une liste qui s'appelle liste **\$liste_pdo**. Puis les données seront mises à la position qui correspond (voir exemple ci-dessous). La position qui correspond est déterminée en fonction du nom du tableau et du nombre de requêtes pour ce tableau.

```

-$liste_pdo['tableaux1'][0]-

```

Puis le programme php parcourt par ligne les données obtenues en les mettant en forme. Ensuite le programme met les données récupérées dans une liste qui s'appelle `$donnee_contenu_tableaux[]`. A la fin de la ligne le programme utilise la fonction de la class tableaux qui est : `creerligne($donnee_contenu_tableaux)`. Pour différencier le header d'une ligne normale, quand je crée une nouvelle ligne je mets comme paramètre 0 ou 1, sachant que 0 signifie une ligne d'en-tête et que 1 c'est une ligne quelconque.

Voici la fonction de la classe Tableaux `creerligne()` :

```
function creerligne($contenu_ligne){  
    //on créer une ligne du tableaux(le header n'est pas pris en compte)  
    $this->lignetotal[]=new Ligne($contenu_ligne,1);  
}
```

Lignetotal est une liste de type object qui appartient à la class Tableaux. Son objectif est de créer un object de la classe ligne, cet object on le stocke dans cette liste ligetotal.

Voici le constructeur de la class Ligne :

```
class Ligne{  
    //liste pour les contenu d'une ligne  
    public $contenu = array();  
    //boolean pour savoir si c'est le header 0 ou les donnee 1  
    public $nbval;  
    function __construct($contenu_ligne,$nbval){  
        $this->nbval=$nbval;  
        //recupere le ligne du row de la requette sql  
        for($i=0;$i<count($contenu_ligne);$i++){  
            $this->contenu[]=$contenu_ligne[$i];  
        }  
    }  
}
```


Dans le constructeur, on récupère la ligne et on la met dans une variable publique de la class Ligne. Cette méthode me sert pour garder en mémoire les lignes du tableau correspondant.

Puis à la fin on construit le tableau en appelant la fonction d'affichage de la classe tableaux. La fonction d'affichage va parcourir tous les objets ligne qu'on a créés pour après afficher les lignes comme tableaux avec la balise tr.

Fonction d'affichage de la classe tableaux :

```
function affichage($liste_critique){
    //recupere la position de signalement critique, le max , le min et le numero du tableaux
    $position = $liste_critique[0];
    $max = $liste_critique[1];
    $min = $liste_critique[2];
    $numtable = $liste_critique[3];
    //affiche le header
    echo "<thead>";
    echo "<tr class='title'>";
    //appelle la fonction d'affichage pour la ligne header
    $this->lignetotal[0]->affichage($position,$max,$min,$numtable);
    echo "</tr>";
    echo "</thead>";
    //affiche les donnee
    echo "<tbody>";
    //boucle sur la liste de type objet de la classe Ligne
    for($i=1;$i<count($this->lignetotal);$i++){
        echo "<tr class='text'>";
        //appelle les fonction d'affichage pour la class Ligne
        $this->lignetotal[$i]->affichage($position,$max,$min,$numtable);
        echo "</tr>";
    }
    echo "</tbody>";
}
```

Lignetotal est une liste de type objet, elle contient tous les objets de la classe ligne. Donc les lignes du tableau

Les balisage du tableaux

Le principe est d'afficher le header en premier avec la balise **<thead>** puis après d'afficher toutes les autres lignes dans une balise qui s'appelle **<tbody>**. Le **<tr>** nous sert pour dire que c'est une ligne du tableau. Le **<tbody>** et le **<thead>** servent à différencier les titres du contenu. La fonction qu'on appelle est une fonction qui appartient à la classe Ligne et sert à afficher les cellules **<td>** des lignes du tableau.

	Overall		On disk		On tape		disk/overall ratio		Avg file size
adonis	2 845 GIB	538 309 files	2 845 GIB	538 309 files	0 GIB	0 files	100.0%	100.0% (files)	5.4 MIB
ams	2 149 022 GIB	1 870 989 files	33 307 GIB	585 264 files	2 115 715 GIB	1 285 725 files	1.5%	31.3% (files)	1 176.2 MIB
antares	3 925 351 GIB	11 224 223 files	86 841 GIB	157 966 files	3 838 510 GIB	11 066 257 files	2.2%	1.4% (files)	358.1 MIB
archivage	246 277 GIB	5 842 631 files	152 873 GIB	5 830 744 files	93 404 GIB	11 887 files	62.1%	99.8% (files)	43.2 MIB
auger	668 080 GIB	16 890 174 files	157 084 GIB	15 610 754 files	510 996 GIB	1 279 420 files	23.5%	92.4% (files)	40.5 MIB
babar	2 175 254 GIB	1 682 477 files	97 GIB	14 files	2 175 157 GIB	1 682 463 files	0.0%	0.0% (files)	1 323.9 MIB

Ligne <tr>

cellule <td>

head <thead>

4. Réflexions et conclusion

J'ai eu la chance de faire un stage passionnant. Ce stage a été très enrichissant, que ce soit sur le monde du travail ou sur la technique de développement (par exemple json). Le stage m'a aussi confirmé la très grande différence entre le monde du travail et le monde des études supérieures. Le stage m'a aussi permis de confirmer mon envie de faire du développement web dans mon travail professionnel.

5. Annexe : Certificat de stage

Lycée Edouard BRANLY 25, rue de Tourvielle 69005 LYON 04 72 16 70 00	BTS Systèmes Numériques Option A Informatique et Réseaux	Session 2023
STAGE EN MILIEU PROFESSIONNEL		
CERTIFICAT DE STAGE <i>(à faire figurer dans le rapport de stage)</i>		

Nom et prénom du stagiaire : Jundt-Schmitter Oscar

Raison sociale de l'entreprise : Centre de Calcul de l'IN2P3

Nom du responsable de l'entreprise : Pierre-Etienne Macchi

Service d'accueil du stagiaire : Groupe Support

Nom du tuteur : Kachelhoffer Thomas Fonction : Ingénieur informaticien

N° de tél. : 04.72.69.41.77

Mèl : kachelho@cc.in2p3.fr

Dates de début et de fin du stage : 23/05/2022 au 01/07/2022

Nombre de ½ journée(s) d'absence excusée(s) : 1 non excusée(s) :

Activités conduites par le stagiaire pendant le stage :

Reprise et développement d'une application web permettant le monitoring de l'utilisation des ressources informatiques du Centre de Calcul par ses utilisateurs (physiciens de l'IN2P3-CNRS). Ce site dispose d'une visibilité interne et externe. Cette activité s'est traduite par :

- Implémentation d'un nouveau driver MySQL dans l'ensemble du code écrit en PHP.
- Réorganisation complète du code avec écriture et mise en commun de nouvelles fonctions et classes en PHP, écriture de nouveaux scripts PHP en intégrant de nouvelles requêtes SQL.
- Mise en œuvre et structuration de fichiers de description/configuration des scripts PHP au format JSON.

Appréciation générale du tuteur sur le stagiaire :

Oscar a fait preuve de curiosité et s'est rapidement approprié le projet. Il a su faire des propositions argumentées concernant la structuration et l'architecture du code. Ses connaissances techniques concernant les langages et technologies utilisés dans le projet sont indéniables et il a su acquérir et compléter le peu qui lui manquait. Oscar a également su faire preuve d'une autonomie certaine dans son travail et appliquer les directives qui lui ont été données. Il a donné entière satisfaction.

Signature et cachet de l'entreprise  Christèle PROT-DU-LAS Responsable informatique Centre de Calcul de l'IN2P3	 CC - IN2P3 Centre de Calcul Institut National de Physique Nucléaire 11 de Physique des Particules 21 Av. Pierre de Coubertin CS70202 91892 7 Villeurbanne Cedex	
---	--	---

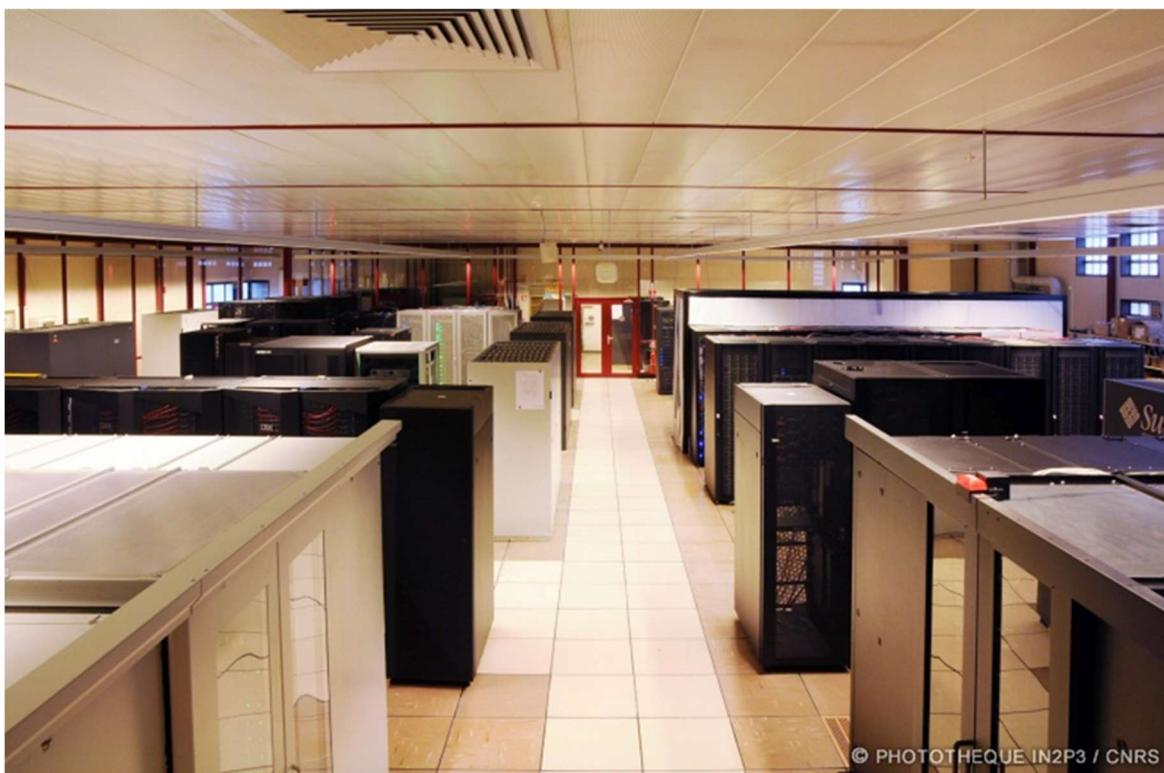
6. Annexe : Document Anglais

7. Annexe : autre

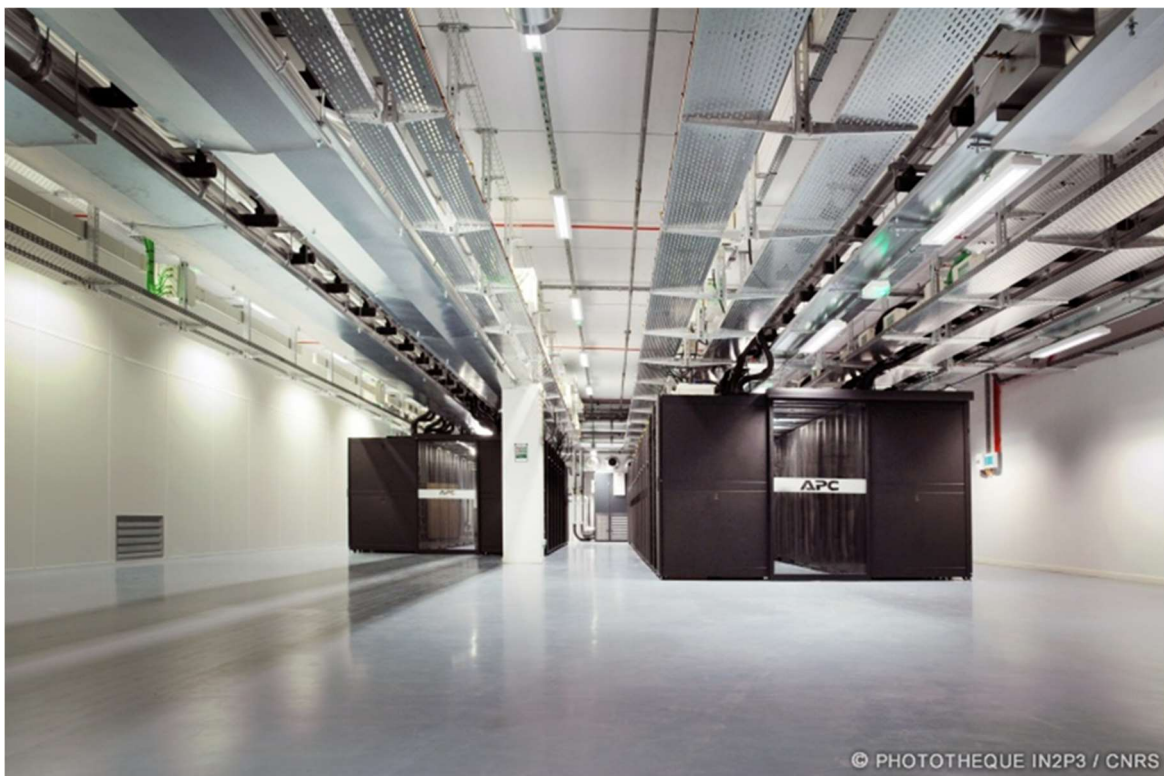
7.1. PHOTO SALLE SERVEUR

Voici quelques photos des deux salles serveurs du CC-IN2P3.

7.1.2. PHOTO SALLE SERVEUR 1



7.1.3.PHOTO SALLE SERVEUR 2



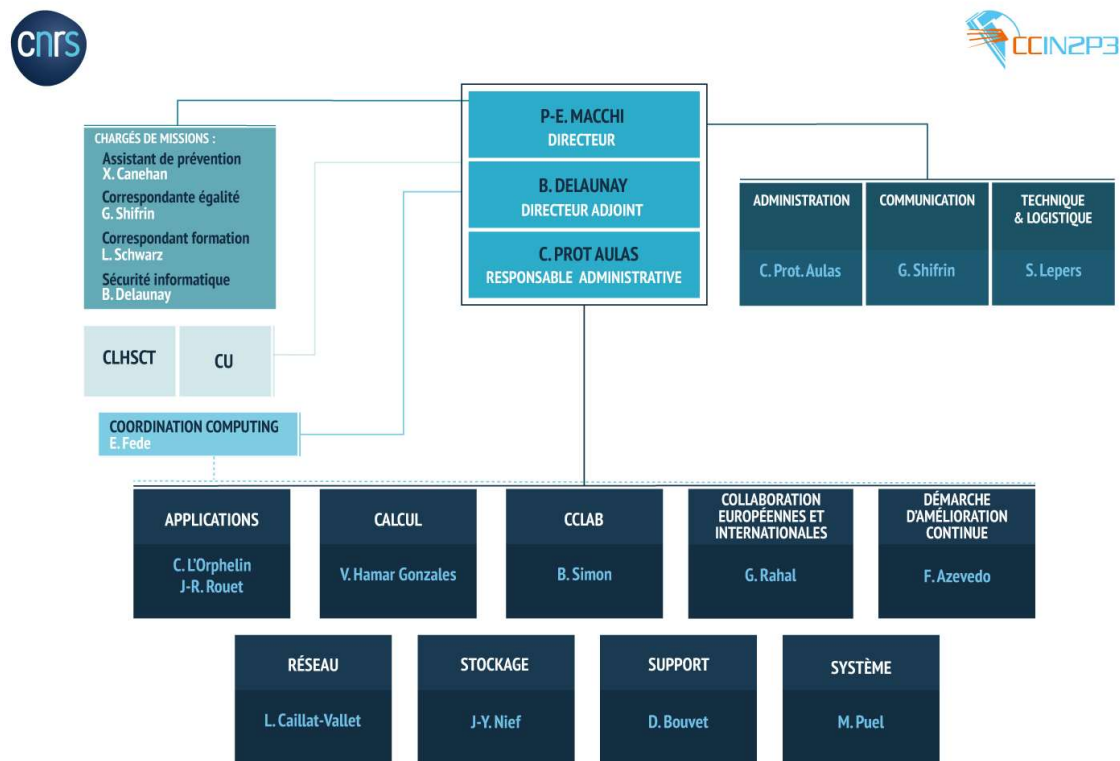
7.2.PHOTO CARTE



7.3.LES ORGANIGRAMMES

Voici les différents organigrammes du Centre.

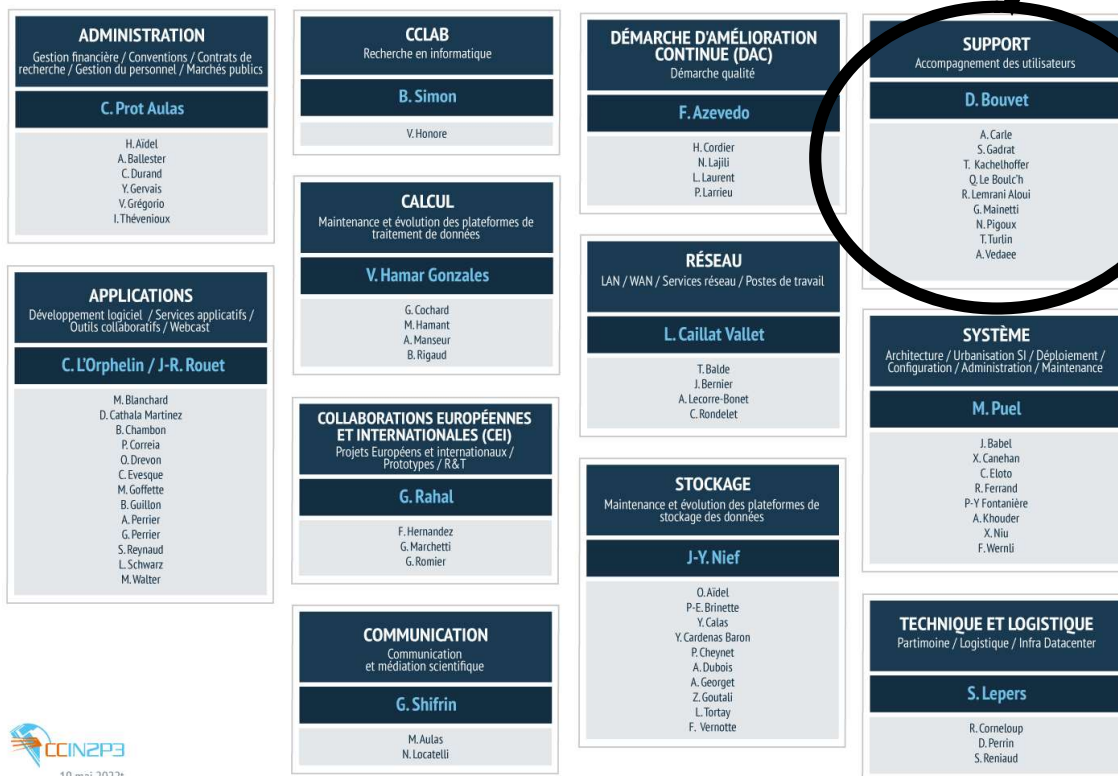
7.3.1.ORGANIGRAMME (POUR CC-IN2P3) GENERAL



Centre de Calcul de l'IN2P3 / CNRS - 21 avenue Pierre de Coubertin - CS70202 - 69627 VILLEURBANNE cedex - FRANCE

19 mai 2022

7.3.2. ORGANIGRAMME (POUR CC-IN2P3) PRECIS



7.4.LIENS UTILISES POUR LE PROJET

(cour de développement en ligne)<https://www.w3schools.com/>

(code externe utiliser)<https://neil.fraser.name/software/tablesort/>

(le site web développer)<https://cctools.IN2P3.fr/mrtguser/oscar/>

(webmail de l'in2p3)<https://zimbra.IN2P3.fr/#2>

(la doc officiel de php)<https://www.php.net/manual/fr/intro-whatism.php>

(la doc de mozilla
developpeur)<https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON>