



MONITORITZACIÓ DE RENDIMENT I ACCOUNTING DE CENTRES DE COMPUTACIÓ GRID

Memòria del Projecte Fi de Carrera
d'Enginyeria Informàtica realitzat per

Ignasi Sans Claret

i dirigit per

Diego Javier Mostaccio Mancini

Universitat Autònoma de Barcelona

Bellaterra, 17 de Juny de 2010



El sotasignat, Diego Javier Mostaccio Mancini
Professor de la Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció
per en Ignasi Sans i Claret.

I per tal que consti firma la present.

Signat: Diego Javier Mostaccio Mancini
Bellaterra, 22 de Juny de 2010

El sotasignat, Francisco Martínez , de l'empresa PIC (*Port d'Informació Científica*)

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat en l'empresa sota la seva supervisió mitjançant un conveni entre PIC (*Port d'Informació Científica*) i DACSO (*Departament d'Arquitectura de Computadors i Sistemes Operatius*) de la Universitat Autònoma de Barcelona.

Així mateix, l'empresa en té coneixement i dóna el vist-i-plau al contingut que es detalla en aquesta memòria.

Francisco Martínez

Bellaterra, 22 de Juny de 2010

Campus UAB Edifici D – 08193 Bellaterra – Barcelona – Spain – Tel +34 93 581 41 09 – Fax +34 93 581 41 10

<http://www.pic.es>

Índex

1	Introducció	7
1.1	Objectius del Projecte	8
1.2	Estructura de la memòria	9
2	Estat de l'Art	10
2.1	Què és el PIC	10
2.2	Serveis del PIC	13
2.2.1	dCache	13
2.2.2	Enstore	14
2.2.3	FTS (file transfer service)	14
2.2.4	LFC (LCG File Catalog)	15
2.2.5	PBS (Portable Batch System)	15
2.3	Sistemes de monitoratge actuals	16
3	Anàlisi del Projecte	18
3.1	Requeriments funcionals	18
3.2	Requeriments tècnics	20
3.2.1	Col·lectors	20
3.2.2	Base de dades	21
3.2.3	Interfície Gràfica	22
3.3	Recursos materials	24
3.4	Planificació temporal del treball	25
3.5	Viabilitat del projecte i riscos	25

4	Disseny	29
4.1	Mòduls del programa	29
4.1.1	Gestor de col·lectors	29
4.1.2	Col·lector Enstore	30
4.1.3	Col·lector dCache	30
4.1.4	Col·lector LFC	31
4.1.5	Col·lector FTS	31
4.1.6	Col·lector DDD(Oracle)	32
4.1.7	Col·lector PBS	32
4.1.8	Base de dades	33
4.1.9	GWT(Google Web Toolkit)	34
4.1.10	Interfície web	34
4.2	Comunicacions dels mòduls	35
4.2.1	Gestor de Col·lectors → Col·lectors	35
4.2.2	Col·lectors → Base de dades de servei	35
4.2.3	Col·lectors → Base de dades d'accounting central	35
4.2.4	Interfície web (Client) → Servidor web (GWT)	36
4.2.5	Servidor web (GWT) → Base de dades d'accounting central	36
4.2.6	Base de dades d'accounting central → Servidor web (GWT)	36
4.2.7	Servidor web (GWT) → Interfície web (Client)	36
4.3	Base de dades	37
5	Implementació del Projecte	38
5.1	Funcionalitat del Programa	38
5.2	Tasques del projecte	40
5.2.1	Configuració de la màquina d'accounting	40
5.2.2	Implementació de la base de dades de accounting	41
5.2.3	Implementació del col·lector de Enstore	42
5.2.4	Implementació del servidor d'aplicacions i la pàgina web (GWT)	42
5.2.5	Afegir aplicació/widget de plotting	45
5.2.6	Testeig del plotting de Enstore	46

ÍNDEX	6
5.2.7 Implementació de col·lector i testeig de LFC i FTS	46
5.2.8 Implementació del col·lector i testeig dCache	47
5.2.9 Implementació del col·lector i testeig de PBS	47
5.2.10 Testeig de la aplicació completa	48
5.2.11 Millores de rendiment	48
5.2.12 Millores de interfície gràfica	48
5.3 Mètodes de implementació	49
5.3.1 Col·lectors	49
5.3.2 Base de dades	49
5.3.3 Interfície d'usuari	49
6 Proves i resultats	50
6.1 Proves	50
6.1.1 Col·lectors	50
6.1.2 Base de dades	51
6.1.3 Interfície	52
6.2 Resultats	53
7 Conclusions i treball futur	54
8 Annex 1	60

Capítol 1

Introducció

El projecte consisteix en la creació de un sistema de monitoratge¹ de determinats serveis que gestiona el Port de Informació Científica[1] aquest sistema de monitoreig estarà principalment compost per programes col·lectors de dades, una base de dades i una interfície web. Aquests serveis generen una informació de accounting² en formats diversos, des de bases de dades PostgreSQL o Oracle fins a fitxers de log en format de text, cadascun dels col·lectors³ treballarà de manera similar, però la seva implementació serà diferent degut a les diferències en el sistema d'emmagatzematge que utilitzi cada servei. La importància de que aquests serveis estiguin en correcte funcionament les 24h del dia fa necessari un sistema de monitoratge per poder estar al corrent en tot moment de l'estat dels sistemes. Concretament els serveis que requereixen d'aquest monitoratge són dCache, Enstore, PBS, LFC, FTS i DDD (Oracle). Al apartat 2.2 d'aquesta memòria s'hi pot trobar una breu explicació de la funció de cadascun d'ells.

L'objectiu del projecte és recopilar la informació de accounting generada, homogeneïtzar-la, desar-la en una base de dades central de accounting i consultar-la

¹Observar per mitjà d'aparells especials el curs de un o vàris paràmetres d'algun tipus de naturalesa per detectar possibles anomalies.

²Accounting: (Comptabilitat) En aquest cas, fa referència al sistema de mesures adoptat per a mesurar el funcionament dels serveis. (Temps en execució, tasa d'errors, estat actual, etc.)

³Programa que recopila les dades d'accounting de una estructura de dades determinada per a desar-les a la base de dades del sistema de monitoreig.

mitjançant una aplicació web per a mostrar-la gràficament d'una forma interactiva i entenedora.

1.1 Objectius del Projecte

- **Anàlisi de Informació disponible:** Anàlisi exhaustiva de les diferents fonts de dades amb les que es treballarà, s'ha de veure quins camps en comú presenten o si a cada servei li correspondran taules diferents de la base de dades. També s'ha de decidir quina informació del accounting actual dels serveis resulta realment útil per al projecte.
- **Construcció de la Base de dades de accounting:** Amb la informació recopilada en la fase anterior s'ha de fer un disseny de la base de dades sobre la qual treballarà la futura aplicació web i en la que els col·lectors hi bolcaran la informació de accounting recollida dels diferents serveis.
- **Programació dels col·lectors:** Els col·lectors seran scripts i s'encarregaran de les tasques necessàries per traslladar la informació de accounting des de la base de dades o fitxer on es trobi fins a la base de dades de accounting. Mitjançant aquest pas es farà un filtratge de la informació i normalitzaran les dades.
- **Estudi de les Mètriques d'interès:** S'han d'estudiar les mètriques existents en els diferents sistemes de monitoratge dels que ja disposa el PIC i afegir les que siguin necessàries per part dels futurs usuaris de la aplicació.
- **Programació de l'aplicació web final per al plotting de les mètriques:** La aplicació web final mostrarà els valors obtinguts de les mètriques gràficament. Això s'aconseguirà gràcies a les diferents eines existents actualment. Algunes de les que es disposa son graph Tools, Google Web Toolkit, que ofereix solucions molt atractives i interactives. S'ha d'estudiar quina es la opció més bona i que ofereix uns gràfics més útils per a la interpretació de les mètriques.

1.2 Estructura de la memòria

A continuació d'aquest capítol de la memòria es pot trobar en el capítol 2 l'estat de l'art del projecte, que ajuda a situar-se en el context en el que es troba el projecte, després de explicar què és el PIC, es procedeix s'explica en què consisteixen els serveis dels quals es pretén fer el monitoreig i com funciona actualment el sistema de monitoreig del PIC.

L'apartat d'anàlisi en el capítol 3, parla dels requeriment que te el projecte i l'anàlisi de les solucions possibles que es podien adoptar en els diferents casos així com les justificacions de per què s'ha utilitzat una tecnologia i no una altre.

En el capítol 4, Disseny, es pretén donar una visió del projecte en forma de blocs que es comuniquen entre si, descrivint el funcionament general dels blocs i les comunicacions que tenen uns i altres.

L'apartat de implementació al capítol 5 explica com s'ha procedit en els diferents aspectes i es concreta a nivell més tècnic algunes de les tasques que s'han hagut de dur a terme en el transcurs del projecte.

Finalment al capítol 6 en els resultats es fa un recull de les proves que s'han efectuat sobre el software desenvolupat, així com els paràmetres que s'han tingut més en compte per decidir si un resultat era l'esperat o no i es mostra en forma de captures de pantalla la interfície web.

Al apartat de conclusions, capítol 7, es fa un repas dels objectius que s'han pogut assolir i els que no, així com un petit anàlisi de per què no s'han assolit. També es plantegen opcions de desenvolupament del software posterior i possibles camins a seguir.

Capítol 2

Estat de l'Art

2.1 Què és el PIC

El Port d'Informació Científica, va ser fundat al Juny de 2003 i es mantingut a través de la coordinació de les següents quatre institucions:

- Departament d'Educació i Universitats (DEiU)
- Centro de Investigaciones Energéticas, Medioambientales y Tecnológica (CIEMAT)
- Universitat Autònoma de Barcelona (UAB)
- Institut de Física d'Altes Energies (IFAE)

És un centre de dades de excel·lència per al processat de informació científica, i ofereix els seus serveis per a grups de científics que treballen en projectes que requereixen molts recursos de computació per l'anàlisi de grans quantitats de dades distribuïdes.

Des de la seva creació al 2003, el PIC ha col·laborat estretament amb el Grid de Computació del LHC [2] (Large Hadron Collider) com a centre Tier-1 i coordina la participació del les set institucions de investigació (CIEMAT, IFAE, IFCA, IFIC, UAM, UB i USC) que constitueixen la contribució espanyola al LCG (LHC Computing Grid).

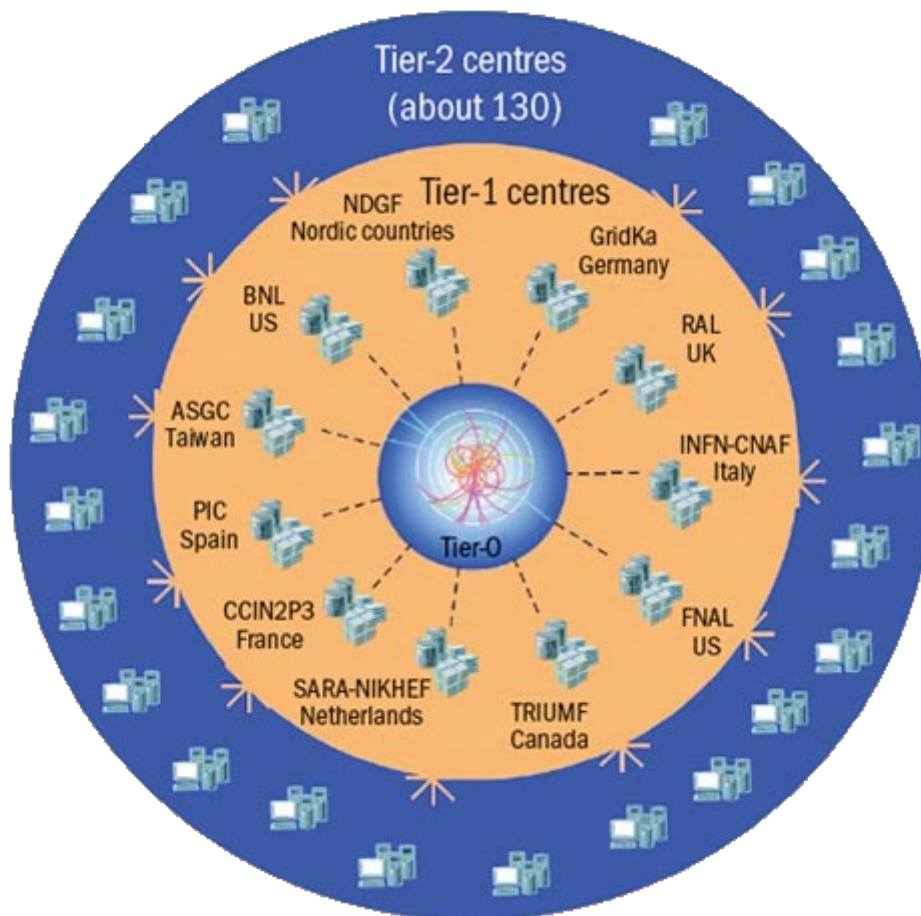


Figura 2.1: LHC Computing Grid

El Port d'Informació Científica és:

- Un factor clau per projectes que requereixen l'emmagatzemament i processat de grans quantitats de dades en equips col·laboratius i sota condicions extremadament difícils.
- Un proveïdor de tecnologia, innovació, coordinació i economies de escala.
- Un centre de desenvolupament de tècniques i serveis generals en el context del emergent GRID de Informació Global.
- Un centre de excel·lència per permetre a Espanya participar en els projectes Europeus per al desenvolupament internacional de la infraestructura del GRID Internacional per a la ciència i tecnologia.

L'estructura del PIC es divideix en diferents departaments, com es pot veure en la figura 2.2, cada un amb unes funcions específiques. Així, trobem el departament de Projectes del LHC on els investigadors i científics del LHC treballen amb els recursos del PIC en projectes directament relacionats amb el Accelerador de Partícules de Ginebra. També hi ha el departament de projectes de neuro, en el qual hi treballen neuropsicòlogues portant projectes de investigació mèdica de l'Hospital de Sant Pau. El departament de Infraestructures s'encarrega del funcionament i instal·lació de nou hardware a la sala de màquines, gestiona el correcte funcionament de les instal·lacions a un nivell més físic i directe. I el departament de Serveis, en el qual s'ha realitzat aquest projecte hi treballen enginyers informàtics encarregats de garantir el correcte funcionament en tot moment dels serveis del PIC, ja sigui el gestor de espai de disc virtual dCache o el gestor de las cues de jobs del PIC, PBS. Dins de cada departament es requereix un seguiment del rendiment i ús dels seus recursos, d'aquí la importància que té el monitoreig i accounting constant i en temps el més real possible depenent de les circumstàncies de cada cas.

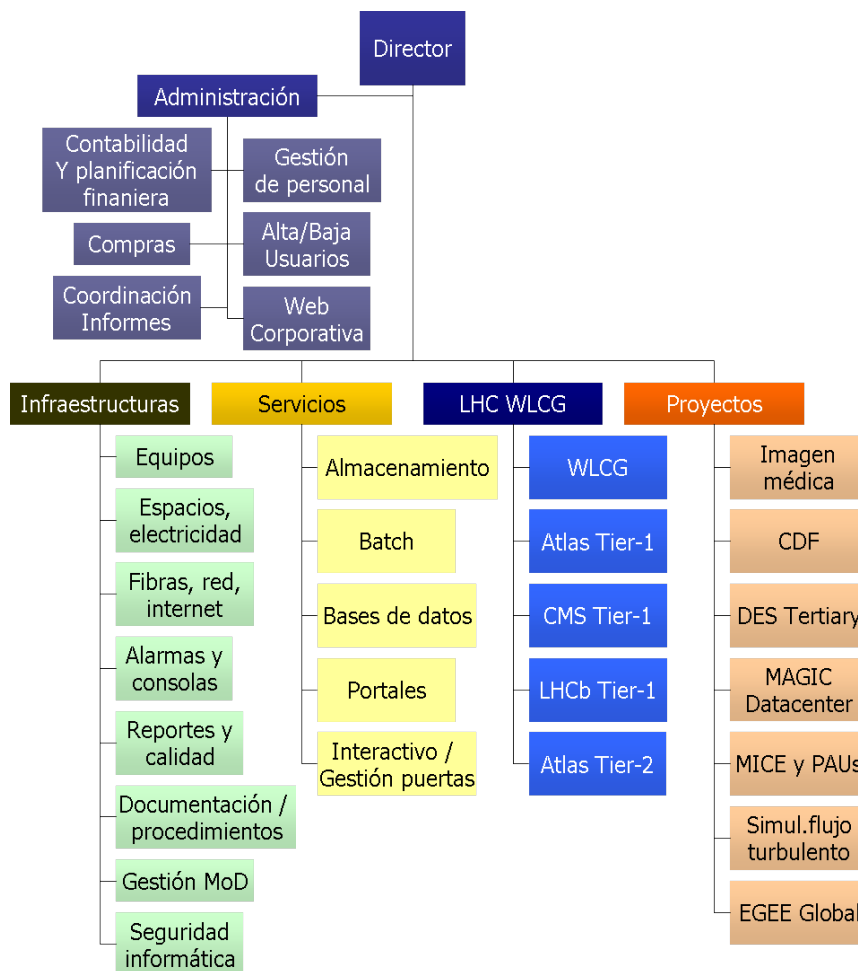


Figura 2.2: Organigrama del PIC

2.2 Serveis del PIC

2.2.1 dCache

Sistema per emmagatzemar i recuperar grans quantitats de informació, distribuïda en una gran quantitat de nodes servidors, sota un sol sistema de fitxers virtual amb una gran varietat de mètodes d'accés. Depenent del Model de Persistència, dCache[3] proporciona mètodes per intercanviar informació amb sistemes d'emmagatzematge backend i per gestionar l'espai, replicació de sets de dades, determinació de "hot spots" i recuperació de fallides de disc o nodes. Connectat a un sistema d'emmagatzematge terciari, la cache simula accés directe il·limitat al espai d'emma-

gatzematge. Els intercanvis de dades cap i des dels HSM subjacents es realitzen de forma automàtica i invisible per l'usuari. Les operacions d'espai de noms del sistema de fitxers es poden realitzar a través d'una interfície NFS (2) estàndard.

El accounting de dCache consisteix en l'emmagatzematge de les dades de interès generades per el servei en una base de dades PostgreSQL.

2.2.2 Enstore

Enstore es el servei que s'encarrega directament de controlar el emmagatzematge en cintes. El PIC disposa de cintes de diferent mides, des de 200 fins a 800 GB, hi ha dos robots per moure aquestes cintes, aquests robots són braços mecanitzats que troben la cinta en el sistema i la llegeixen. Les comandes de consulta de dades de les cintes es van posant en cues fins que n'hi ha suficient per fer una consulta, degut a que la consulta a una cinta requereix mes temps que un consulta normal a disc, per el temps que triga el robot en fer el desplaçament en cada consulta, però es llegeix i grava un volum d'informació més gran. Les operacions amb les cintes han de ser amb fitxers grans o amb quantitats d'informació gran dis una mateixa cinta per a resultar eficients, per això existeix el sistema de cues.

El accounting de Enstore consisteix en l'emmagatzematge de les dades de interès generades per el servei en una base de dades PostgreSQL.

2.2.3 FTS (file transfer service)

Crea canals virtuals de comunicació entre centres de investigació a diferents punts del territori. Aquests canals es creen i destrueixen segons la necessitat dels investigadors. Així en un moment determinat es pot requerir un canal entre dos centres de Tier-1 o amb el Tier-0(Cern) mateix.

El accounting de FTS consisteix en l'emmagatzematge de les dades de interès generades per el servei en una base de dades Oracle i fitxers de log.

2.2.4 LFC (LCG File Catalog)

És un catàleg que conté un mapeig de fitxers, de situació lògica a física de les dades de que es disposa en les cintes/discs. Serveix per trobar fitxers sol·licitats per a fer una transferència dins del disc/cinta sense necessitat de treballar-hi directament, permetent treballar a un nivell d'abstracció més elevat. En el LFC un fitxer determinat està representat per un GUID (GRID Unique Identifier). Gràcies a això, un fitxer replicat a diferents sites es considera que és el mateix fitxer i apareix com a una entrada única dins del catàleg lògic del LFC.

En aquest servei és útil tenir un seguiment de les connexions en un instant de temps entre diferents centres, les lectures i escriptures concurrents o els esborrats de fitxers i dades en general. De totes aquestes operacions concretament interessa saber quines tenen èxit i quines no.

El *accounting* de LFC consisteix en l'emmagatzematge de les dades de interès generades per el servei en una base de dades Oracle.

2.2.5 PBS (Portable Batch System)

És un gestor de cues, s'encarrega d'assignar els jobs que han arribat del Grid de Computació del LHC a les màquines del PIC que els executaran. Genera informació de *accounting* d'interès per als investigadors com SE de procedència, usuari, temps d'execució, temps de cua (*idle*). L'inconvenient principal que hi ha en l'actual sistema de monitoreig és que un mateix job genera varies línies del log depenent del estat en que es trobi (*Queued*, *Active*, *Exited*), fins i tot un mateix job que comenci un dia i acabi un altre dia diferent, deixarà informació d'*accounting* en diversos logs. Principalment es requereix fer un seguiment del temps d'execució dels jobs que han esta llençats al Grid.

El *accounting* de PBS consisteix en l'emmagatzematge de les dades de interès generades per el servei en uns fitxers de log.

2.3 Sistemes de monitoratge actuals

Actualment al PIC es disposa d'alguns mecanismes de monitoratge que s'han anat desenvolupant com a solucions temporals per a saber el estat dels serveis de forma mes o menys acurada. Aquestes aplicacions treballen directament sobre les dades de accounting generades per els diferents serveis però no tots ells disposen del seu sistema de monitoratge, alguns com PBS simplement generen dades de monitoreig a fitxers de log, els quals han de ser consultats manualment per el responsable del servei.

Un servei sense monitoratge requereix que el usuari interessat en conèixer el seu estat tingui coneixements avançats de com està fet el accounting del servei ja sigui el disseny de la base de dades, l'estructura dels logs, etc. I no permet extreure resultats vistosos com gràfics que permetin veure el funcionament del servei de forma rapida i entenedora.

Alguns serveis disposen actualment de monitoratge mitjançant diferents pàgines web, però aquests sistemes són independents els uns dels altres i han estat creats sense uns criteris i pautes comuns de manera que no permeten el mateix visionat de la informació ni un seguiment global dels serveis, i depenen molt de la que fos la necessitat de la persona que les va crear en el seu moment i han de ser gestionats personalment per els responsables dels serveis.

Enstore genera dades d'accountig que son desades a una base de dades PostgreSQL, i mitjançant graph tools es mostra el seu estat actual a partir de metriques mitjançant una interfície web similar a la que es preten implementar amb el projecte.

Dcache també desa les dades generades a una base de dades PostgreSQL i les mètriques són mostrades de forma més rudimentaria a la web.

LFC i FTS desen les seves dades de accounting a una base de dades Oracle, i el estat dels canals virtuals es visible desde un entorn web.

DDD (Oracle) genera dades de accounting i les desa en una base de dades Oracle, però no disposa de sistema de monitoreig.

PBS genera fitxers de log, i els desa a disc. Per a saber el estat de PBS s'ha de consultar aquests fitxers manualment.

Capítol 3

Anàlisi del Projecte

3.1 Requeriments funcionals

El projecte amb el que ens trobem s'ha plantejat des d'un començament amb requeriments oberts. Degut a que les mètriques podran anar canviant durant el transcurs del desenvolupament de l'aplicació, no és possible deixar-ho tancat en un principi, ni tan sols en el final del projecte quedaran tancades. La aplicació en si mateixa ha de quedar preparada per acceptar noves mètriques en un futur, noves formes de visualitzar les dades d'accounting. Malgrat això, hi ha una sèrie de requeriments funcionals essencials que sí que es poden definir al inici del projecte.

A continuació s'exposen els requeriments funcionals essencials del projecte.

- **Veure informació de accounting dels Serveis** La informació de accounting dels diferents serveis dels quals es desitja fer un seguiment s'ha de trobar localitzada en una única base de dades. La intenció, és reunir la informació actual d'accounting que es troba repartida en múltiples bases de dades i en diverses màquines en una única base de dades central d'accounting i en cas que sigui possible combinar i relacionar aquesta informació. D'aquesta manera es disposarà de la informació d'accounting d'interès centralitzada i homogeneïtzada.

- **Definir les mètriques inicials** Inicialment s'han de definir una sèrie de mètriques per part dels administradors dels serveis, amb aquestes es partirà com a base des de la qual treballar.
- **Monitoratge dels Serveis** A través de les mètriques que es defineixin, els investigadors han de poder fer un seguiment del estat dels serveis del PIC de manera ràpida i entenedora. El monitoratge ha de consistir en una sèrie de gràfiques de les mètriques que seran visualitzades a través de una interfície web.
- **Visualització gràfica de les mètriques** Les mètriques hauran de ser visualitzades a través de la pàgina web de forma clara i entenedora, de manera que tant els administradors dels serveis com la resta d'usuaris en puguin fer una ràpida interpretació.
- **Períodes de temps variable per a les gràfiques mostrades. Definibles per al usuari final** Els resultats mostrats al web s'han de poder modelar des de la interfície de la aplicació web mateix, de manera que si es vol veure un període de temps més gran del que es mostra inicialment, sigui fàcil i ràpid demanar-ho a la aplicació.
- **Diferents estils de gràfica** L'usuari del web ha de poder canviar el estil de la gràfica, i aquesta ha de ser configurable des de la pàgina web mateix (Quality plot, pie chart, bar, stacked bar). D'aquesta manera, s'han de poder visualitzar els resultats de les mètriques de diferents maneres, sempre i quan es tracti de visualitzacions possibles. Un conjunt de dades que es visualitzi en format de Stacked bar, no es pot traduir a format de Pie chart.

3.2 Requeriments tècnics

3.2.1 Col·lectors

Els col·lectors han de ser programes que no requereixin de l'ús de molts recursos, han de connectar-se a bases de dades, fer consultes, tractar els volums d'informació recopilats, i fer insercions a la base de dades central.

Per a la programació dels col·lectors es preferible treballar amb un llenguatge basat en scripting que no requereixi una compilació a ser possible, al mateix temps això facilitarà una futura modificació d'aquests per a adaptar-los a canvis en les bases de dades d'accounting dels serveis. Els col·lectors han de ser programes lleugers i ràpids, per això des de un començament només s'ha plantejat la opció de programarlos en aquest tipus de llenguatges.

Dins d'aquest àmbit, hem contemplat la opció de programar en llenguatge Python és un llenguatge de programació d'alt nivell de propòsit general. Guido van Rossum el va crear el 1991. Combina una potència remarcable amb una sintaxi clara i entenedora. Python és comparat amb altres llenguatges com Java, Tcl, Perl o Scheme. Utilitza sagnats com a delimitadors de blocs, fet poc freqüent als llenguatges de programació. Una altra de les característiques d'aquest llenguatge és el tipificat dinàmic i la capacitat per interpretar el codi en temps d'execució, en contradicció d'altres llenguatges com ara C, que ho fan en temps de compilació. Python ens ofereix l'avantatge de no requerir una compilació, aquest fet resulta atractiu per al nostre propòsit, i al mateix temps es un llenguatge estructurat basat en classes, anomenats mòduls en aquest cas.

En l'elaboració dels col·lectors no es requereix tanta complexitat, i per això s'ha contemplat la opció de treballar en Shell de Linux, ja que degut a que totes les màquines amb les que es treballa funcionen sobre distribucions de Linux i aquestes ja porten incorporat l'interpret d'ordres de Shell, ni tan sols s'ha de instal·lar software addicional.

La Shell és un fitxer executable que ha d'interpretar les ordres, transmetre'ls al sistema i llançar el resultat. Hi ha diversos shells. La més comú és sh (anomenada Bourne shell), bash (Bourne again Shell), csh (C Shell), tcsh (Tenex C shell), ksh (Korn shell) i zsh (zero shell). Generalment, els seus noms coincideixen amb el nom de l'executable.

Shell actua com un intermediari entre el sistema operatiu i l'usuari gràcies a línies d'ordres que aquest darrer introdueix. La seva funció és la de llegir la línia d'ordres, interpretar el seu significat, dur a terme la comanda i després llançar el resultat per mitjà de les sortides.

La versió de Shell utilitzada hauria de ser Bourne Shell que és la més comú, per garantir la compatibilitat amb tots els sistemes Linux amb els que es treballi.

3.2.2 Base de dades

La base de dades que ha de concentrar la informació d'accounting dels diferents serveis ha de ser estable i robusta, però no es requereix de un Sistema Gestor de Bases de Dades molt complexe, les dades contingudes no han de ser modificades, ja que es tracta de dades d'accounting, temps de lectura d'un disc, canals creats des del PIC a un altre centre, jobs que han començat en un moment determinat, etc.

En el camp de Gestors de Bases de Dades Relacionals s'ha de tenir en compte les diferents possibilitats que ofereix el mercat actual, en el PIC es treballa bàsicament amb Oracle i PostgreSQL, però també hi ha altres opcions com Microsoft SQL Server de Microsoft i altres RDBMS amb llicència lliure com MySql o Firebird.

Oracle és un sistema de gestió de base de dades relacional (o RDBMS per l'acrònim en anglès de Relational Data Base Management System), desenvolupat per Oracle Corporation. Es considera a Oracle com un dels sistemes de bases de dades més complets i robusts del mercat, però per contra té uns costos de llicències molt elevats.

PostgreSQL és un sistema de gestió de base de dades relacional orientat a objectes i lliure, publicat sota la llicència BSD. A diferència de Oracle, el desenvolupament de PostgreSQL no és manejat per una sola empresa sinó que és dirigit per una comunitat de desenvolupadors i organitzacions comercials les quals treballen en el seu desenvolupament.

L'avantatge principal de PostgreSQL sobre Oracle és el seu cost, PostgreSQL és gratuït i s'utilitza en el PIC cada cop més, de fet la política del PIC és utilitzar PostgreSQL en la major mesura del possible, es va plantejar la opció de treballar amb PostgreSQL o Oracle, però la preferència des del PIC és clara, per això el RDBMS escollit per a aquest projecte ha estat PostgreSQL.

3.2.3 Interfície Gràfica

La interfície gràfica del projecte ha de ser una pàgina web que mostri resultats de consultes a la base de dades gràficament. Per això, és necessari un servidor web que treballi sobre la base de dades com a intermediari de la interfície web que ha d'utilitzar l'usuari final.

Per a aquest propòsit, s'ha optat per a desenvolupar aquest entorn mitjançant una eina de Google, GWT(Google Web Toolkit), que permet la fàcil integració de aplicacions web creades a pàgines web així com la utilització de widgets.

GWT o Google Web Toolkit és un framework creat per Google que permet amagar la complexitat de diversos aspectes de la tecnologia AJAX. És compatible amb diversos navegadors, la qual cosa és notòria ja que cada navegador sol necessitar codi específic per aconseguir un front-end correcte en una aplicació web. El concepte de Google Web Toolkit és bastant senzill, bàsicament el que s'ha de fer és crear el codi en Java utilitzant qualsevol entorn de desenvolupament (IDE)¹ de Java i el compilador ho traduirà a HTML i JavaScript.

¹Integrated Development Environment. Normalment consisteix en un editor de codi font, un compilador o intèrpret, eines d'automatització i un debugador.

a source code editor a compiler and/or an interpreter build automation tools a debugger

Amb la biblioteca GWT, els desenvolupadors poden crear i depurar aplicacions AJAX en llenguatge JAVA utilitzant l'entorn de desenvolupament que prefereixin. Quan una aplicació és desplegada, el compilador GWT tradueix l'aplicació Java a un arxiu Javascript. GWT no és només una interfície de programació, proporciona un conjunt d'eines que permeten desenvolupar funcionalitats Javascript d'alt rendiment en el navegador del client.

Es poden depurar aplicacions AJAX en un IDE com es faria amb una aplicació d'escriptori, i en un navegador qualsevol igual que es faria si estigués codi JavaScript. El plug-in de desenvolupador GWT abasta la bretxa entre el bytecode de Java en el depurador i JavaScript del navegador.

Es pot utilitzar el mateix cicle d'edició-actualització-vista en la que es sol treballar a amb el llenguatge JavaScript, mentre que al mateix temps, es poden inspeccionar variables, establir punts d'interrupció, i utilitzar totes les eines de depuració que es disposa amb Java.

La construcció d'aplicacions AJAX d'aquesta manera és més productiu gràcies a un major nivell d'abstracció per sobre de conceptes comuns com la manipulació del DOM i la comunicació XHR. Així, una persona acostumada a treballar amb JAVA pot treballar còmodament sense canviar de llenguatge però per a acabar aconseguin codi JavaScript optimitzat per el Engine de Google.

S'ha escollit GWT per a treballar per que es una solució innovadora i atractiva que agradava ja des de un començament al PIC, i així es va demanar que fos tingut en compte.

3.3 Recursos materials

Per al desenvolupament de les diferents parts del projecte es disposa de un DELL PowerEdge 400sc amb un sistema operatiu Linux Ubuntu 9.04. Aquesta màquina permet treballar ja des de un primer moment amb el intèrpret d'ordres Shell i crear els Shell Scripts que funcionaran de col·lectors de la base de dades.

El projecte es realitzarà amb el sistema operatiu Ubuntu Linux, i es treballarà des de l'aplicació Eclipse[5] que permet portar un control exhaustiu de tots els fitxers de codi i de documentació, a més de donar eines per a facilitar la programació.

A la estació de treball utilitzada se li ha instal·lat un entorn de treball Eclipse per al desenvolupament de la aplicació web. GWT disposa de les eines per a ser integrat a Eclipse, i això, junt amb el fet que és un software conegut i que s'ha utilitzat durant la carrera ha ajudat a la decisió de treballar amb aquest IDE des de un principi.

També s'ha instal·lat un client i un servidor de PostgreSQL per fer proves locals sense necessitat de treballar amb la màquina que ha de fer de servidor de la web i la base de dades definitius.

Per a la base de dades es disposa de una màquina virtual dedicada especialment per al desenvolupament d'aquest projecte facilitada des del PIC. Aquesta màquina té 2 cores, 8 gigabytes de memòria RAM i 50 gigabytes de disc. La màquina virtual ha de contenir la base de dades i se li han instal·lat un client i un servidor de PostgreSQL, la memòria RAM i el disc disponible són suficients per al projecte i per a treballar amb comoditat amb la base de dades al mateix temps que els col·lectors.

3.4 Planificació temporal del treball

Planificació inicial de les tasques que es duran a terme durant el transcurs del projecte, esta subjecte a canvis ja siguin de base com replantejar la manera de desar les dades de accounting o de disseny, a l'hora de decidir com es mostraran els resultats a la aplicació web final.

A les figures 3.1 i 3.2 al final d'aquest capítol es pot veure com queda aquesta planificació.

3.5 Viabilitat del projecte i riscos

En aquest punt s'estudien els riscos que impedirien el desenvolupament del projecte o interromprien el seu avanç, i també si es considera que és viable.

- **Temps insuficient** S'hauria de retallar alguns aspectes del projecte i fer-lo menys ambiciós, deixar-lo a punt però per a ser ampliat més endavant treballant de forma modular des de un començament perquè la ampliació sigui el més fàcil possible.
- **Pèrdua de les dades de accounting** En aquest cas seria necessari treballar amb dades dummy per aconseguir visualitzar resultats, fins a tenir noves dades. Si treballem en mode "read only" amb les dades originals no hauríem de tenir aquest problema. I en el PIC es disposa de còpies de seguretat de tota la informació d'accounting emmagatzemades en màquines diferents, per tant aquest cas seria molt poc probable.
- **Fallides en els servidors del PIC** Seria molt complicat treballar els col·lectors sense tenir els serveis disponibles, simplement es podrien dissenyar els algorismes mentre duressin les fallides. Afortunadament des del PIC hi ha un control exhaustiu de la disponibilitat dels seus serveis en tot moment i una fallida general es molt improbable. Per a no haver de interrompre el desenvolupament del projecte, es treballa amb còpies dels programes en la estació de treball. I es poden realitzar proves amb la base de dades local.

- **Dades errònies en el accounting original** Si per alguna raó les dades de les quals disposem no son correctes, els resultats mostrats tampoc ho seran, fet que podria desencadenar a pensar que el que no es correcte es el programa o col·lectors utilitzats. Aquesta situació és improbable, però per a estar segurs sempre es pot comptar amb el suport dels administradors dels serveis per a verificar la veracitat de la informació.

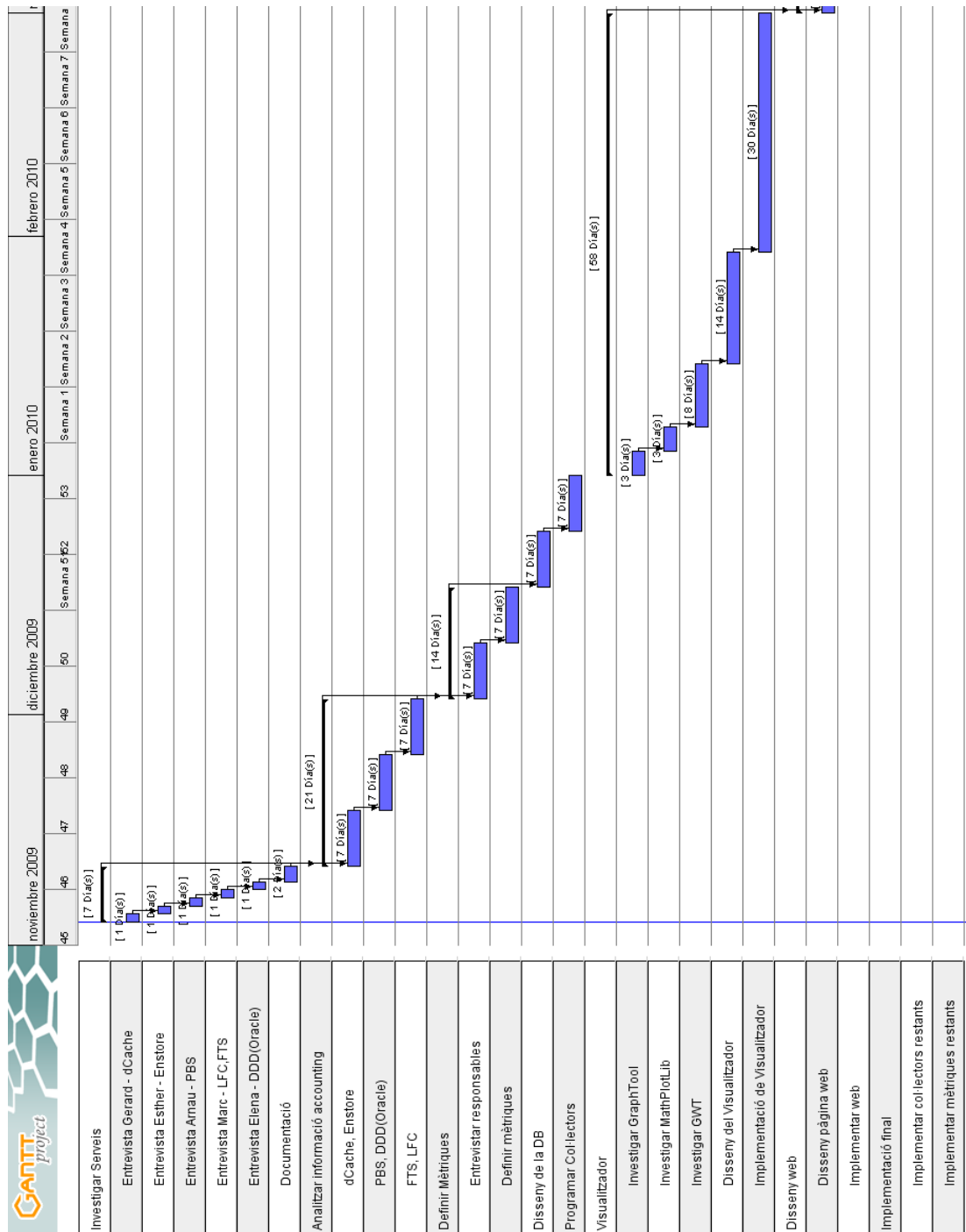


Figura 3.1: Planificació inicial (Part 1)

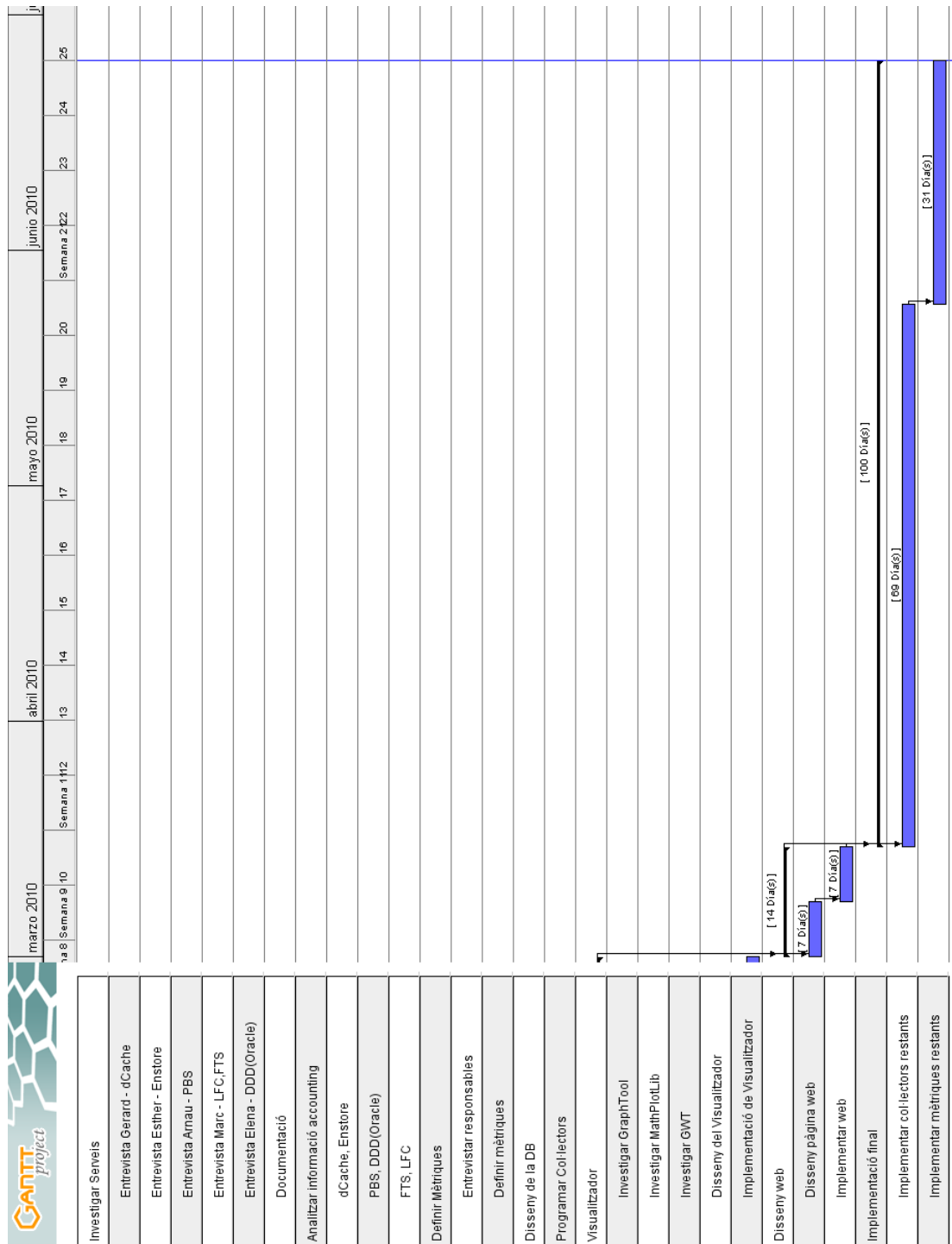


Figura 3.2: Planificació inicial (Part 2)

Capítol 4

Disseny

En aquest apartat es contemplen els diferents mòduls amb els que comptarà el projecte i la interacció que tindran els uns amb els altres.

En un primer punt es parla de cada un dels mòduls especificant quin tipus de mòdul és, quines dades poden entrar al mòdul per a la seva interpretació o processat, quines surten cap a un altre mòdul i el comportament associat al mòdul en qüestió.

4.1 Mòduls del programa

4.1.1 Gestor de col·lectors

- **Tipus:** Script Bourne Shell de tipus daemon.
- **Entra:** -
- **Surt:** -
- **Comportament:** Script que s'executarà cada 5 minuts, llençarà els col·lectors dels serveis per adquirir les noves dades d'accounting.

4.1.2 Col·lector Enstore

- **Tipus:** Script Bourne Shell
- **Entra:** Informació d'accounting procedent de la maquina d'accounting de Enstore. Dades noves des de la última crida al col·lector.
- **Surt:** Informació d'accounting nova de Enstore dirigida a la base de dades centralitzada. Inserció de les noves dades a la base de dades d'accounting central.
- **Comportament:** En cas de no tenir dades de les taules de Enstore a la base de dades d'accounting central, aquest script fa un bolcat de les taules senceres per a omplir amb totes les dades que puguin interessar. Si ja hi havia informació, el que fa el script es consultar de quan són les dades més recents que tenim i fes una consulta a la base de dades de Enstore per a obtenir la informació nova des d'aquell punt. Figura 4.1.

4.1.3 Col·lector dCache

- **Tipus:** Script Bourne Shell
- **Entra:** Informació d'accounting procedent de la maquina d'accounting de dCache. Dades noves des de la última crida al col·lector.
- **Surt:** Informació d'accounting nova de dCache dirigida a la base de dades centralitzada. Inserció de les noves dades a la base de dades d'accounting central.
- **Comportament:** En cas de no tenir dades de les taules de dCache a la base de dades d'accounting central, aquest script fa un bolcat de les taules senceres per a omplir amb totes les dades que puguin interessar. Si ja hi havia informació, el que fa el script es consultar de quan són les dades més recents que tenim i fes una consulta a la base de dades de dCache per a obtenir la informació nova des d'aquell punt.

4.1.4 Col·lector LFC

- **Tipus:** Script Bourne Shell
- **Entra:** Informació d'accounting procedent de la maquina d'accounting de LFC. Dades noves des de la última crida al col·lector.
- **Surt:** Informació d'accounting nova de LFC dirigida a la base de dades centralitzada. Inserció de les noves dades a la base de dades d'accounting central.
- **Comportament:** En cas de no tenir dades de les taules de LFC a la base de dades d'accounting central, aquest script fa un bolcat de les taules senceres per a omplir amb totes les dades que puguin interessar. Si ja hi havia informació, el que fa el script es consultar de quan són les dades més recents que tenim i fes una consulta a la base de dades de LFC per a obtenir la informació nova des d'aquell punt.

4.1.5 Col·lector FTS

- **Tipus:** Script Bourne Shell
- **Entra:** Informació d'accounting procedent de la maquina d'accounting de FTS. Dades noves des de la última crida al col·lector.
- **Surt:** Informació d'accounting nova de FTS dirigida a la base de dades centralitzada. Inserció de les noves dades a la base de dades d'accounting central.
- **Comportament:** En cas de no tenir dades de les taules de FTS a la base de dades d'accounting central, aquest script fa un bolcat de les taules senceres per a omplir amb totes les dades que puguin interessar. Si ja hi havia informació, el que fa el script es consultar de quan són les dades més recents que tenim i fes una consulta a la base de dades de FTS per a obtenir la informació nova des d'aquell punt.

4.1.6 Col·lector DDD(Oracle)

- **Tipus:** Script Bourne Shell
- **Entra:** Informació d'accounting procedent de la maquina d'accounting de DDD(Oracle). Dades noves des de la última crida al col·lector.
- **Surt:** Informació d'accounting nova de DDD(Oracle) dirigida a la base de dades centralitzada. Inserció de les noves dades a la base de dades d'accounting central.
- **Comportament:** En cas de no tenir dades de les taules de DDD(Oracle) a la base de dades d'accounting central, aquest script fa un bolcat de les taules senceres per a omplir amb totes les dades que puguin interessar. Si ja hi havia informació, el que fa el script es consultar de quan són les dades més recents que tenim i fes una consulta a la base de dades de DDD(Oracle) per a obtenir la informació nova des d'aquell punt.

4.1.7 Col·lector PBS

- **Tipus:** Script Bourne Shell
- **Entra:** Informació d'accounting procedent de la maquina d'accounting de PBS.
- **Surt:** Informació d'accounting nova de PBS dirigida a la base de dades centralitzada.
- **Comportament:** Script que analitza la informació d'accounting de PBS. Com que PBS treballa amb fitxers de logs, el funcionament serà diferent a la resta d'scripts. Aquí el que fara el script serà comprovar que hi hagi fitxers nous a la màquina de accounting de PBS i en cas de haver-hi fitxers nous, analitzar-los per al tractament de les dades. S'ha d'extreure la informació nova de accounting i actualitzar la informació ja existent.

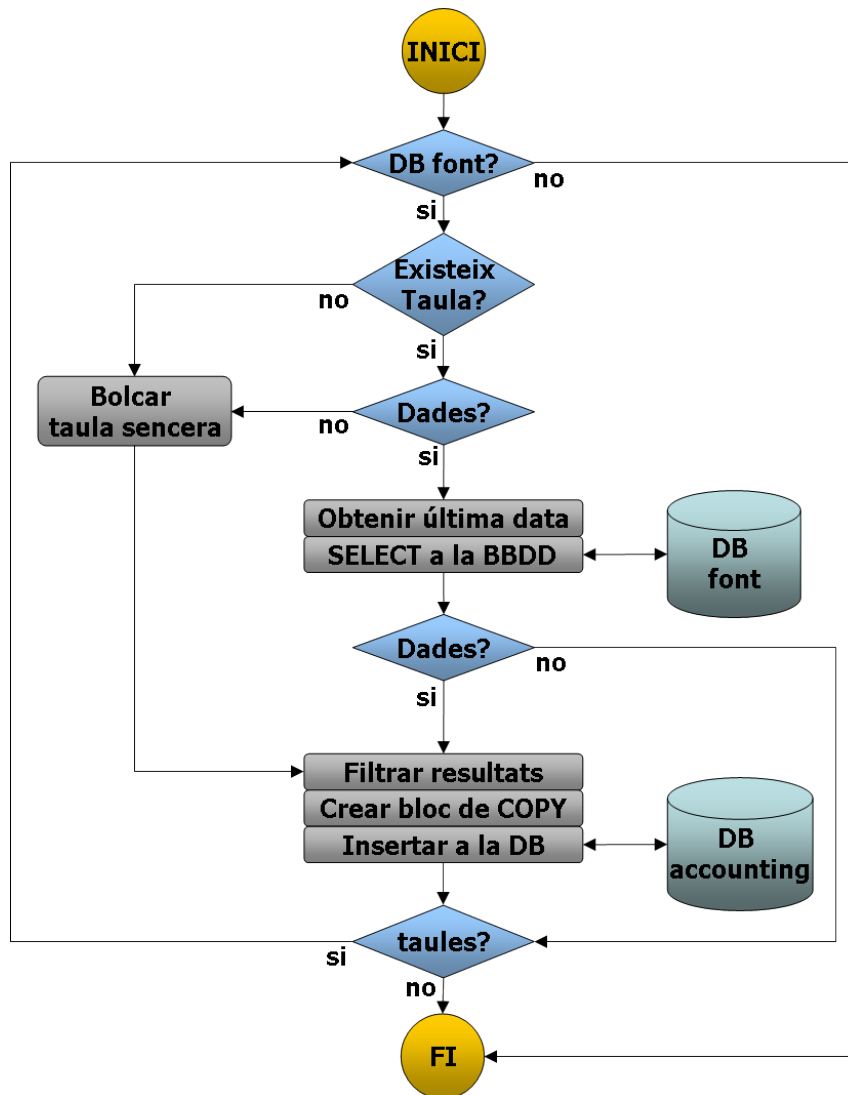


Figura 4.1: Diagrama de fluxe de un Col·lector

4.1.8 Base de dades

- **Tipus:** Base de dades PostgreSQL
- **Entra:** Informació d'accounting dels diferents serveis del PIC procedent dels col·lectors.
- **Surt:** Resultats de les consultes SQL del servidor d'aplicacions web.
- **Comportament:** La base de dades d'accounting rebrà INSERTs procedents dels col·lectors i SELECTS procedents de l'aplicació web.

4.1.9 GWT(Google Web Toolkit)

- **Tipus:** Gestor de servlets
- **Entra:** Informació retornada per la base de dades, peticions de les aplicacions llençades per els usuaris.
- **Surt:** Pàgina web de plotting resultant de compilar JSP, resultats de les aplicacions.
- **Comportament:** El servidor d'aplicacions estarà esperant peticions de la pagina web per a fer les consultes pertinents a la base de dades d'accounting central.

4.1.10 Interfície web

- **Tipus:** Pàgina web HTML+Javascript
- **Entra:** Informació retornada per la base de dades, peticions de les aplicacions llençades per els usuaris.
- **Surt:** Pàgina web de plotting resultant de compilar JSP, resultats de les aplicacions.
- **Comportament:** El servidor d'aplicacions estarà esperant peticions de la pagina web per a fer les consultes pertinents a la base de dades d'accounting central.

4.2 Comunicacions dels mòduls

En aquest apartat s'estudien les comunicacions existents entre els diferents mòduls del sistema, en la figura 4.2 es poden veure aquestes relacions gràficament.

4.2.1 Gestor de Col·lectors → Col·lectors

Crida del sistema. El gestor de col·lectors fa una crida als col·lectors cada cinc minuts per recopilar la nova informació d'accounting, la resta de temps resta en espera. El gestor de col·lectors ha de ser un programa de tipus daemon, sempre ha d'estar en execució en espera de tornar a ser cridat.

4.2.2 Col·lectors → Base de dades de servei

Els col·lectors després de ser cridats per el gestor de col·lectors es comunicaran amb la base de dades corresponent al servei del qual s'ocupen. Així, el col·lector de Enstore, consultarà les taules de les quals n'extreu dades i consultarà les dades noves des de l'última crida, per a insertar-les posteriorment a la base de dades central d'accounting.

4.2.3 Col·lectors → Base de dades d'accounting central

Els col·lectors s'han de comunicar amb la base de dades d'accounting central per a obtenir la informació necessària per saber de quan són les dades més recents que hi ha a la base de dades mitjançant noves insercions. A continuació procedeixen amb la comunicació anteriorment explicada.

Un cop els col·lectors tenen les noves dades de les seves respectives fonts, ja siguin bases de dades o fitxers de log, procedeixen a comunicar-se novament amb la base de dades d'accounting central per a fer-hi les noves insercions mitjançant la comanda COPY de PostgreSQL que permet inserir blocs de dades inserint files senceres, a diferència del INSERT tradicional que és més lent.

4.2.4 Interfície web (Client) → Servidor web (GWT)

A la pàgina web un usuari pot escollir una mètrica per visualitzar el estat del servei. En aquest moment es produeix la comunicació des del client web al servidor web per a obtenir les dades necessàries per a generar el gràfic de la consulta.

De la mateixa manera, un cop la web ja esta mostrant el gràfic anterior, el usuari pot canviar els paràmetres que cregui oportuns per a fer una nova consulta de la mateixa mètrica amb valors diferents, canviant l'interval de temps, el projecte, tipus de job, etc.

4.2.5 Servidor web (GWT) → Base de dades d'accounting central

El servidor d'aplicacions executa comandes SELECT de SQL que li han arribat des de la pàgina web contra la base de dades per consultar les dades necessàries per mostrar gràficament les mètriques.

4.2.6 Base de dades d'accounting central → Servidor web (GWT)

La base de dades envia els resultats de les consultes al servidor web i aquest tracta les dades, per donar un format entenedor per a la aplicació de generació de gràfics.

4.2.7 Servidor web (GWT) → Interfície web (Client)

El servidor web, despres de manipular les dades que li ha retornat la base de dades d'accounting central les envia a la pagina web perquè el codi Javascript de la aplicació web generi la gràfica corresponent.

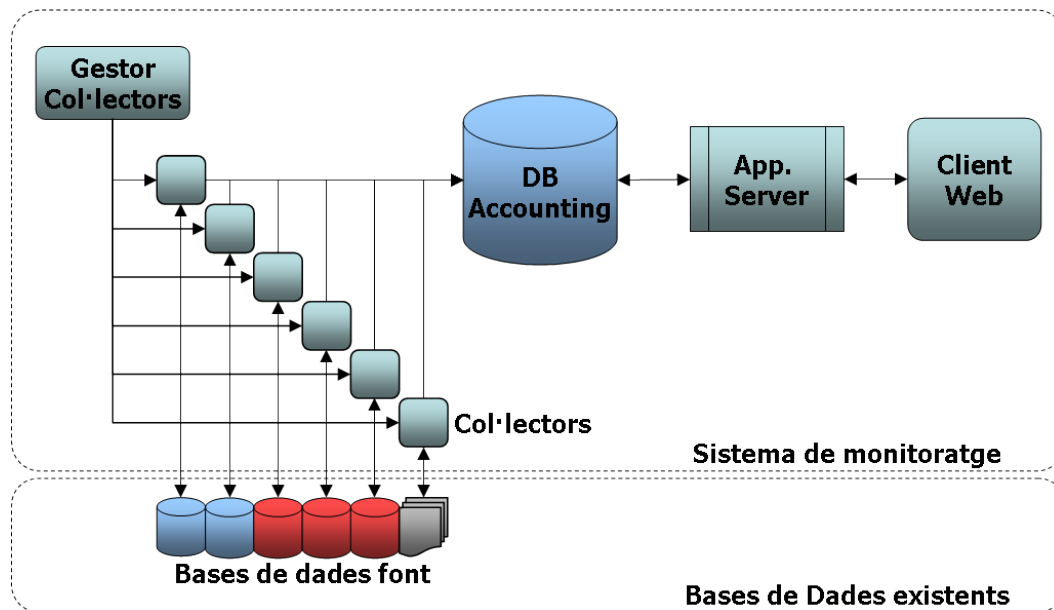


Figura 4.2: Comunicacions entre els diferents mòduls

4.3 Base de dades

El disseny de la base de dades és molt senzill, consisteix en la col·lecció de taules que s'extreuen de les bases de dades d'accounting de cada servei.

Es tracta de una base de dades relacional on basicament es fan insercions i consultes, en algun cas es realitzaran modificacions, però només amb les dades que indiquen el estat de un job provinents del servei de gestió de cues PBS.

Capítol 5

Implementació del Projecte

En aquest apartat s'analitzarà la funcionalitat del programa en termes generals, i es fa un repàs de les tasques en les quals s'ha dividit el projecte i en què consisteixen cada una de elles. S'analitzen els punts més conflictius del desenvolupament del sistema de monitoratge i s'exposen les solucions trobades.

5.1 Funcionalitat del Programa

La aplicació de cara als usuaris finals consisteix en una pàgina web des de la qual es pot veure gràficament el estat dels diferents serveis dels quals es fa el monitoratge. (dCache, Enstore, PBS, LFC, FTS, DDD Oracle)

Aquesta web disposa de una llista de mètriques prèviament definides entre el programador, els administradors dels serveis i membres del departament LHC. A la llista de mètriques inicial se n'hauran de poder sumar de noves, ja que els requeriments per part dels membres del LHC canvien amb el pas del temps.

La aplicació web programada des de la base proporcionada per GWT acabarà treballant sobre un servidor web TOMCAT mitjançant AJAX. Al seleccionar la mètrica a mostrar, es llençarà una petició al servidor web, i aquest consultarà a la base de dades d'accounting central PostgreSQL en la que es troba continguda la informació d'accounting dels serveis.

La informació d'accounting dels serveis es recopila mitjançant uns programes col·lectors implementats en Bourne Shell de Linux. La informació del servei es obtinguda per a cada un d'aquests programes d'una manera diferent, ja que els serveis emmagatzemen aquesta informació de diverses formes.

- **Col·lector 1 (Enstore):** Consulta de dades a BBDD PostgreSQL
- **Col·lector 2 (dCache):** Consulta de dades a BBDD PostgreSQL
- **Col·lector 4 (FTS):** Consulta a BBDD Oracle
- **Col·lector 5 (LFC):** Consulta a BBDD Oracle
- **Col·lector 6 (3D Oracle):** Consulta a BBDD Oracle
- **Col·lector 3 (PBS):** Recopilació de dades des de fitxers de log.

Al recopilar la informació, els col·lectors la guarden a la base de dades d'accounting central PostgreSQL que estarà disponible a la màquina virtual facilitada per el PIC.

En la figura 5.1, es poden veure els diferents components que conformen el sistema d'accounting que s'ha desenvolupat. Separats per regions delimitades amb punts es distingeix entre les fonts de dades que ja existien en els diferents sistemes d'accounting i el sistema d'accounting implementat.

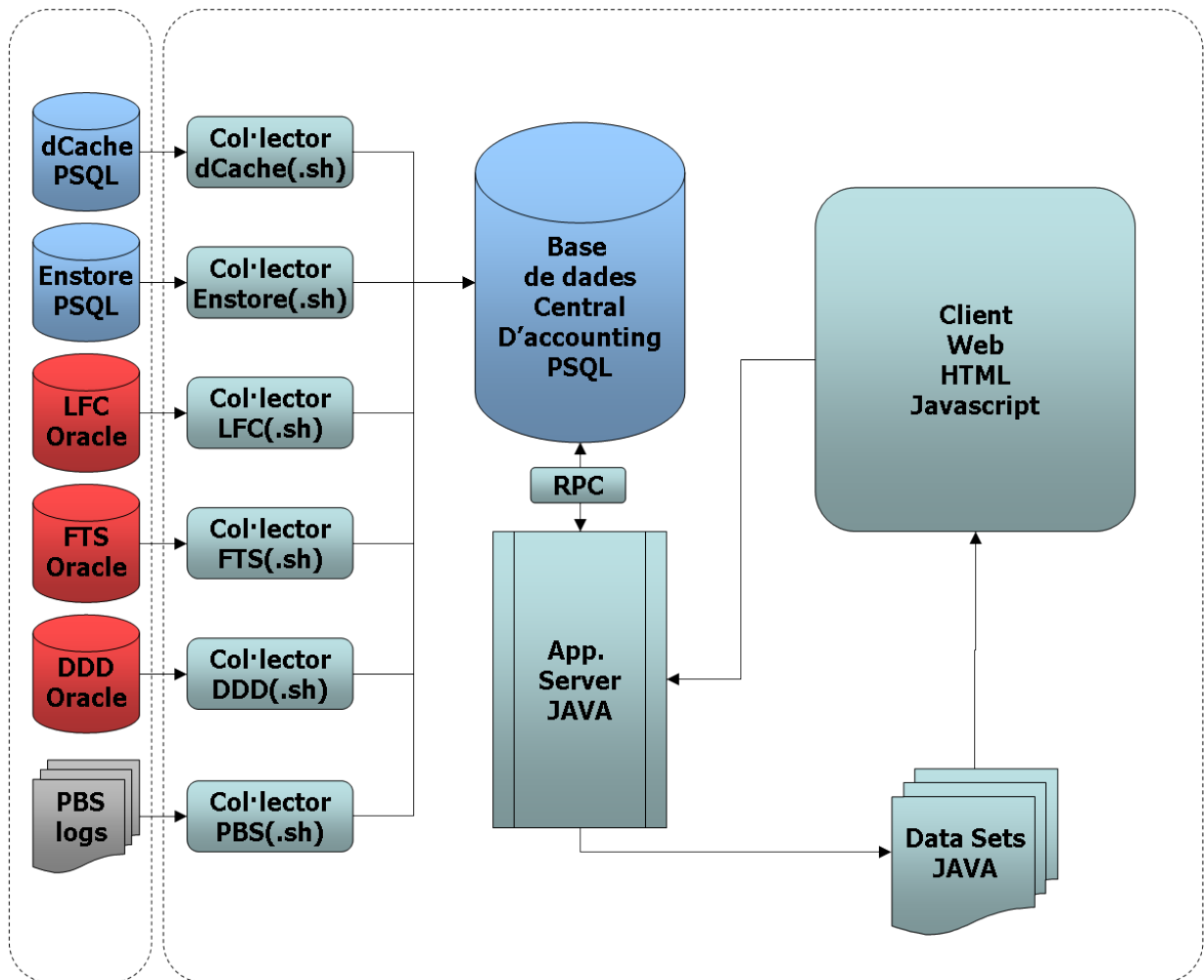


Figura 5.1: Diagrama de blocs del sistema de monitoratge

5.2 Tasques del projecte

S'ha dividit el projecte en les diferents tasques que s'exposen a continuació.

5.2.1 Configuració de la màquina d'accounting

La màquina d'accounting s'ha de configurar per als requisits del projecte. Així, s'ha de garantir l'accés a la màquina per part de les màquines que seran utilitzades per a la realització del projecte. Els accessos es realitzaran utilitzant clients de SSH, per tant, la màquina d'accounting s'ha de dotar de un servidor de SSH i incorporar-hi

les claus d'identificació RSA pròpies de cada màquina que hi ha d'accedir.

A aquesta màquina també és necessari instal·lar-hi la versió de PostgreSQL adequada, per evitar problemes de compatibilitat ha de ser la versió 8.2 que és la utilitzada en els serveis que treballen amb PostgreSQL.

5.2.2 Implementació de la base de dades de accounting

La Base de Dades de Accounting Centralitzat en PostgreSQL admet consultes, insercions, dumps de taules i creació de noves taules en cas de ser necessari. S'ha hagut de configurar el funcionament de la base de dades i l'accés a ella des de màquines externes.

S'ha modificat el fitxer de connexions externes a PostgreSQL propi de la instal·lació de PostgreSQL, i s'ha donat accés a la estació de treball des de la qual es llencen les probes del GWT, així com de la pròpia màquina que conté la base de dades, per a poder fer probes des de la línia de comandes mateix.

L'autenticació del client es controla mitjançant un fitxer de configuració, que tradicionalment s'anomena `pg_hba.conf`¹ i s'emmagatzema en el directori de dades del clúster de bases de dades. Al inicialitzar el cluster de bases de dades de PostgreSQL es crea un arxiu `pg_hba.conf` per defecte que ha de ser modificat per a implementar els canvis esmentats.

També s'ha obert les comunicacions de tipus TCP/IP de PostgreSQL al port estandard de PostgreSQL 5432. Aquesta modificació s'ha realitzat sobre el fitxer de configuració `postgres.conf`.

Es poden realitzar les connexions necessàries dins de la intranet del PIC o des de la VPN si no ens trobem al edifici. La màquina d'accounting no és accessible des de l'exterior, la única manera de accedir-hi és estant connectat a la VPN.

¹HBA significa Host Based Authentication, autenticació basada en host

5.2.3 Implementació del col·lector de Enstore

Mitjançant el llenguatge Bourne Shell i la comanda `pg_dump` de postgres treballa el col·lector de Enstore. En una primera crida aquest col·lector recopila les taules senceres, però en crides posteriors, recopila únicament les dades noves des de la última actualització de la base de dades d'accounting central. En aquestes operacions es produeix la comunicació entre la màquina que conté la base de dades d'accounting central i els col·lectors, i la màquina que realitza el accounting de Enstore.

5.2.4 Implementació del servidor d'aplicacions i la pàgina web (GWT)

Amb la biblioteca GWT, s'han creat i depurat aplicacions AJAX en llenguatge JAVA utilitzant l'entorn de desenvolupament Eclipse. Quan una aplicació és desplegada, el compilador GWT tradueix l'aplicació Java a un arxiu Javascript. GWT proporciona un conjunt d'eines que permeten desenvolupar funcionalitats Javascript d'alt rendiment en el navegador del client sense sortir de la comoditat de treballar amb JAVA.

Per a la comunicació entre la base de dades i el servidor d'aplicacions s'ha hagut d'afegir suport de Postgres i a RPC (Remote Procedure Calls).²

Les consultes a la base de dades d'accounting central són gestionades per un Servlet en la configuració inicial de GWT i es necessari fer trucades RPC, utilitzant un controlador de devolució de trucada a el costat del client per tenir la comunicació amb la base de dades. (és important recordar que l'aplicació GWT s'executa en el client en el navegador, no en el servidor)

Per a poder treballar amb PostgreSQL des de Java, s'han hagut d'instal·lar els

²Crida a Procediment Remot, és un protocol que permet a un programa executar codi en una altra màquina remòtament sense necessitat de haver de preocupar-se de les comunicacions entre un i altre.

drivers necessaris que es poden trobar en el paquet JDBC³. És una API que permet l'execució d'operacions sobre bases de dades des del llenguatge de programació Java, independentment del sistema operatiu on s'executi o de la base de dades a la qual s'accedeix, utilitzant el dialecte SQL del model de base de dades que s'utilitzi. En la figura 5.2 es pot veure la estructura d'aquesta interfície.

Un cop instal·lats els drivers d'accés necessaris tant a la part del client com del servidor que compon la web i el servidor web, el següent pas es crear una classe per a cada tipus de retorn que es vulgui enviar des del costat del servidor al costat del client. En aquest cas el que s'ha fet ha estat utilitzar la classe vector per a contenir els resultats retornats per les consultes a la base de dades, s'ha hagut d'optar per aquest procediment ja que la classe ResultSet amb la que es tracten les dades retornades per les consultes en JAVA no és una classe serialitzable, i la classe Vector sí.

En el context d'emmagatzematge de dades i transmissió, la serialització és el procés de convertir una estructura de dades o un objecte en una seqüència de bits perquè pugui ser emmagatzemat en un arxiu, un buffer de memòria, o transmetre a través d'un enllaç de connexió de xarxa per a ser "ressuscitats" més endavant en el mateix o en un altre entorn informàtic.

És necessari treballar d'aquesta manera amb la base de Dades per a poder obtenir les dades de la base de dades i enviar-les a la aplicació web, on finalment seran utilitzades per a mostrar el gràfic de la mètrica en qüestió.

La part de crear la pàgina web és més senzilla, gràcies al GWT, la web es construeix amb JAVA, així amb la API que ens proporciona google, es pot dissenyar la aplicació amb botons, quadres de text i altres elements web afegint els widgets de google que satisfan les funcions. Aquest codi en JAVA amb widgets de la API de google acaba convertit en Javascript.

³Java DataBase Connectivity[12]

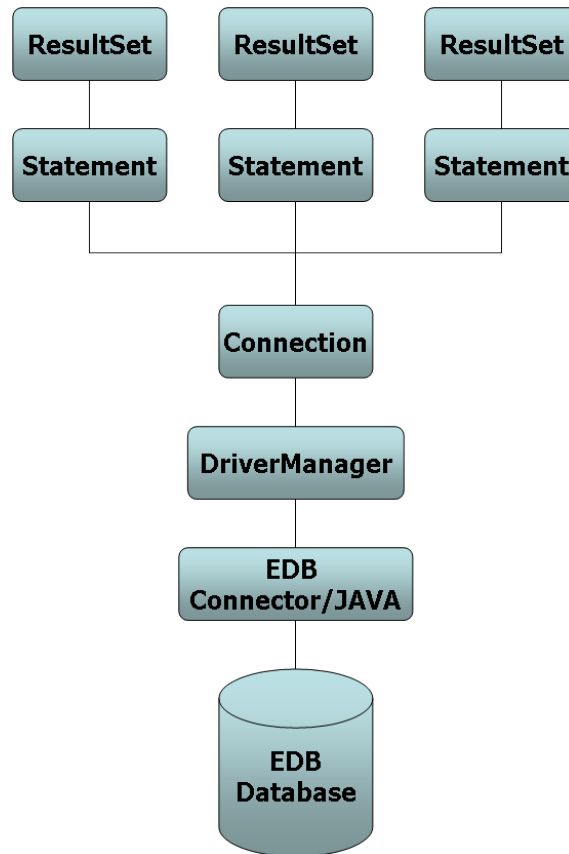


Figura 5.2: Interfície JDBC

A diferència dels minimitzadors de JavaScript, que només funcionen amb text, el compilador de GWT realitza optimitzacions i una anàlisi estàtic complet de tota la base de codi de GWT i, genera codi JavaScript que es carrega i executa amb més rapidesa que el JavaScript equivalent creat de forma manual. Per exemple, el compilador de GWT suprimeix de forma segura tot el codi no utilitzable (mitjançant una exhaustiva tasca d'eliminació de classes, mètodes, camps, i fins i tot paràmetres, que no s'utilitzen) per assegurar-se que l'arxiu de seqüència compilat sigui el més petit possible.

En resum, la pàgina web realitza les consultes de dades al servidor web amb peticions senzilles, el servidor web fa consultes SQL a la base de dades PostgreSQL gràcies al driver JDBC mitjançant Remote Procedure Calls i aquestes dades són desades en objectes serialitzables JAVA per a ser enviats a través de la xarxa a la pàgina web on la aplicació web els podrà utilitzar per a la elaboració de les gràfiques.

A la figura 5.3 es pot veure aquesta interacció gràficament.

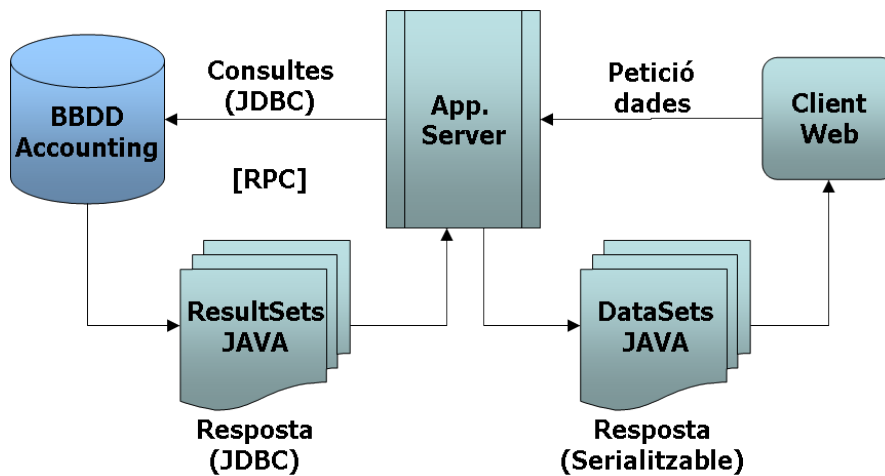


Figura 5.3: Comunicacions entre la Base de Dades d'accounting, el servidor d'aplicacions i la interfície web

5.2.5 Afegir aplicació/widget de plotting

Al nostre servidor de aplicacions afegirem el widget de plotting desenvolupat des de GWT i amb Google chart [7], una senzilla aplicació web que disposa de una API pròpia i ens permet generar imatges en format png de les gràfiques de les quals li donem les dades. Es poden generar gràfiques amb l'avantatge de que la feina de processar les dades per a convertir-les en gràfics recau al servidor de google que s'encarrega de generar-les. A la figura 5.4 es pot veure com des de Google es posa a disposició dels usuaris un espai per a provar els diferents elements de que disposa Google Chart per a dibuixar gràfics.

Google Chart Tools / Image Charts (aka Chart API) [Home](#) [Docs](#) [FAQ](#) [Forum](#) [Terms](#)

Chart API
[What's New?](#)
[Release Notes](#)

Using Charts
[Getting Started](#)
[Data Formats](#)
Live Chart Playground
[Troubleshooting Charts](#)
[POST Requests](#)
[List of Chart Parameters](#)
[Creating a <map> For Your Chart](#)
[User-Submitted Charts](#)

Chart Gallery
[Bar Charts](#)
[Box Charts](#)
[Candlestick Charts](#)
[Compound Charts](#)
[Dynamic Icons](#)
[Formulas](#)
[Google-O-Meter Charts](#)

Live Chart Playground

This page provides a way to test your chart URLs quickly and easily. Paste your URL into the URL text box, and edit the parameters in the Parameters box. The chart and both text boxes update as you type.

Tips:

- Newlines with no indentation are equivalent to "&" in the URL.
- Indented newlines are equivalent to "T" in the URL.
- Click "Reformat Parameters" to clean up the indentation and layout in the parameters box, for ease of editing.
- Autocomplete matches both parameter strings (chxt) and parameter friendly names ("Chart titles") as you type.
- If you use special characters in your label text (such as a literal + sign) you must [URL-encode them](#).

URL: ([Link to this page](#))

```
http://chart.apis.google.com/chart?cht=svg&chs=250x150&chd=s:Monkeys&chxt=x,y&chxs=0,ff0000,12,0,1t|1,0000ff,10,1
```

Parameters:

```
cht=svg
chs=250x150
chd=s:Monkeys
chxt=x,y
chxs=0,ff0000,12,0,1t
      1,0000ff,10,1,1t
```

Generated Chart:

[Show errors...](#)

Figura 5.4: Google Chart Tools Live Chart Playground

L'inconvenient de treballar amb aquest tipus d'eines com en el cas de GWT és que es necessari adaptar-se als formats establerts enlloc de treballar amb formats propis. Afortunadament Google disposa de molta documentació per a facilitar aquest pas.

5.2.6 Testeig del plotting de Enstore

El testeig consisteix a fer el plotting de les mètriques actuals definides a Enstore. Així es pot comprovar si les dades es prenen correctament de la base de dades i els resultats mostrats són els esperats. Aquest testeig implica tots els mòduls de la aplicació ja, des de la base de dades fins a la aplicació web final.

5.2.7 Implementació de col·lector i testeig de LFC i FTS

Mitjançant el llenguatge Bourne Shell treballa el col·lector de LFC i FTS. En una primera crida aquest col·lector recopila les taules senceres, però en crides posteriors, recopila únicament les dades noves des de la última actualització de la base de dades d'accounting central. En aquestes operacions es produeix la comunicació entre la

màquina que conté la base de dades d'accounting central i els col·lectors, i la màquina que realitza el accounting de LFC i FTS.

El testeig consisteix a fer el plotting de les mètriques actuals definides a LFC i FTS. Així es pot comprovar si les dades es prenen correctament de la base de dades i els resultats mostrats són els esperats. Aquest testeig implica tots els mòduls de la aplicació ja, des de la base de dades fins a la aplicació web final.

5.2.8 Implementació del col·lector i testeig dCache

Mitjançant el llenguatge Bourne Shell treballa el col·lector de dCache. En una primera crida aquest col·lector recopila les taules senceres, però en crides posteriors, recopila únicament les dades noves des de la última actualització de la base de dades d'accounting central. En aquestes operacions es produeix la comunicació entre la màquina que conté la base de dades d'accounting central i els col·lectors, i la màquina que realitza el accounting de dCache.

El testeig consisteix a fer el plotting de les mètriques actuals definides a dCache. Així es pot comprovar si les dades es prenen correctament de la base de dades i els resultats mostrats són els esperats. Aquest testeig implica tots els mòduls de la aplicació ja, des de la base de dades fins a la aplicació web final.

5.2.9 Implementació del col·lector i testeig de PBS

El col·lector de PBS ha de ser implementat en llenguatge shell i bàsicament llegeix directament dels fitxers de log en els que es desen les dades d'accounting de PBS. Amb l'ajuda de comandes com el grep es recopila la informació dels fitxers i li se li dona el format adequat per convertir-ho en inserts a la base de dades d'accounting central. En el cas de PBS s'ha de fer un seguiment dels jobs que hi ha llençats en aquest moment i quins han acabat. S'ha de consultar la base de dades abans de fer els inserts per comprovar si es tracta del mateix job però en un esta diferent. S'ha de recordar que els jobs poden ser llençats un dia i acabar al dia següent o més tard

i tot, per tant pot haver un job en diferents estats de execució i en diferents fitxers de log repetit.

El testeig consisteix a fer el plotting de les mètriques actuals definides a PBS. Així es pot comprovar si les dades es prenen correctament de la base de dades i els resultats mostrats són els esperats. Aquest testeig implica tots els mòduls de la aplicació ja, des de la base de dades fins a la aplicació web final.

5.2.10 Testeig de la aplicació completa

El testeig de la aplicació completa consisteix en provar mètriques que impliquin relacions entre taules de diferents serveis, ja que com estan muntats ara mateix els sistemes d'accounting això no és possible. Aquest és un dels objectius del projecte, ens interessa poder interrelacionar els diferents serveis per tenir una visió global del estat del PIC en un moment determinat.

5.2.11 Millores de rendiment

El funcionament del programa no ha d'afectar en el rendiment de les bases de dades d'accounting dels serveis, de manera que la aplicació resulti inocua per al sistema actual d'accounting. El codi és molt poc intrusiu i s'ha procurat que les operacions amb les bases de dades siguin el menys costoses possible, només recuperant informació i adaptant-la a la base de dades d'accounting central mitjançant els scripts col·lectors.

5.2.12 Millores de interfície gràfica

Com a última millora es poden polir aspectes més visuals de l'aplicació, per fer-la més atractiva i donar-li un bon acabat. Aquest detalls únicament visuals es faran en última instància sempre i quan no quedin millores possibles en el apartat de rendiment del programa. Les millores es realitzaran sobre la interfície web del programa.

5.3 Mètodes de implementació

5.3.1 Col·lectors

Programats utilitzant el editor de textos Gedit de Ubuntu en llenguatge Shell. Amb la consola de Ubuntu es proven les comandes per a obtenir resultats immediats. Els manuals de Shell dels que disposa Ubuntu han estat de gran utilitat així com la gran quantitat de pàgines web referents a la consola de Linux que es troben per internet.[8]

5.3.2 Base de dades

La base de dades ha estat implementada mitjançant la línia de comandes de Linux i els repositoris de PostgreSQL[9]. S'ha instal·lat la versió 8.3.11 ja que és compatible amb les versions dels serveis de Enstore i dCache i ens ofereix funcionalitats que no tenim amb versions anteriors, com per exemple noves funcions a utilitzar.

5.3.3 Interfície d'usuari

Per al desenvolupament de la interfície d'usuari i tot el que hi ha darrere, servidor d'aplicacions, pàgina web, comunicacions amb la base de dades... S'ha utilitzat l'entorn de desenvolupament Eclipse i el llenguatge JAVA. El llenguatge de programació venia imposat per GWT, però el fet de que JAVA disposi d'una API[10] tan extensa i ben documentada, així com d'una gran comunitat d'usuaris al darrere, ha fet que es trobessin solucions per als problemes que anaven sorgint.

Capítol 6

Proves i resultats

En aquest capítol s'expliquen algunes proves realitzades sobre els elements del sistema implementat així com proves de rendiment. I a continuació es mostren captures de pantalla de la aplicació final.

6.1 Proves

6.1.1 Col·lectors

Execució. Els col·lectors s'executen i han de generar consultes i insercions a les bases de dades i finalitzar la seva execució fins a la següent crida sense provocar incidents que interrompin la seva execució automàtica cada període de actualització de la base de dades.

Operacions dels col·lectors:

- Consulta de base de dades font
- Inserció de dades a la BD central

Possibles resultats:

- No hi ha dades noves. Al realitzar la consulta a la base de dades font del col·lector, aquesta ha no ha retornat dades noves, per tant no s'haurà de fer cap inserció.

- Obtenim dades. La consulta a la base de dades font del col·lector ha retornat dades noves que hauran de ser inserides a la base de dades. Prèviament el col·lector netejara els resultats de la consulta i els convertirà en una query de inserció mitjançant COPY.
- Base de dades caiguda. La Base de Dades central no està operativa per alguna raó, s'ha de avortar la execució actual i esperar a la propera actualització.

En el desenvolupament dels col·lectors s'han tingut en compte els temps de execució. Es fa un seguiment del temps que s'inverteix en les operacions amb les bases de dades, ja que s'ha de tenir en compte que aquests temps s'han de minimitzar. Les bases de dades d'origen estan subjectes a insercions i consultes constantment, i el rendiment de aquestes no s'ha de veure alterat per la intervenció dels col·lectors.

S'ha comprovat que les operacions de lectura de una taula sencera i inserció són les que més temps consumeixen, per això, s'ha procurat minimitzar-les o en cas de no ser possible evitar-ne el seu ús, que el gruix de dades que utilitzin sigui el més baix possible.

6.1.2 Base de dades

Per verificar que la BD funciona correctament, s'ha comprovat que les consultes realitzades contra la base de dades retornessin els resultats esperats, així com els continguts que calia introduir o eliminar de la base de dades es realitzessin correctament. Aquestes proves s'han realitzat des de diferents màquines i entorns, ja siguin la màquina en la que hi ha el servidor de Postgres i l'entorn de línia de comandes de que disposa, o altres.

S'han realitzat un seguit de proves:

- `postgres@acc.pic.es` → `psqlaccounting`

Des de línia de comandes de la base de dades. Proves de insercions esborrat de taules, esborrat de dades, consultes, en la consola de comandes de que disposa PostgreSQL per a gestionar la base de dades.

- **user@maquinaRemota** → **psql – hacc.pic.es – p5432accounting – cQUERY**

Des de línia de comandes de màquines remotes amb el client de PostgreSQL instal·lat. S'han provat consultes, insercions i esborrats de la base de dades central.

- **JAVA des de Eclipse**

Des de programes java. S'han instal·lat els drivers JDBC necessaris per a crear programes JAVA que amb poques línies de codi realitzin consultes a la base de dades central. Els drivers JDDBS s'utilitzen per treballar amb bases de dades PostgreSQL mitjançant programes JAVA, a la xarxa i a la web de JDBC es poden descarregar i consultar la seva API.

6.1.3 Interfície

Accions per part dels usuaris:

- Seleccionar mètrica.
- Modificar paràmetres de la mètrica i sol·licitar consulta un altre cop.

Resultats possibles:

- La DB esta caiguda. La web retorna un missatge indicant la causa del error. No es pot mostrar la gràfica amb els resultats.
- El servidor web cau. La màquina que allotja la pàgina web cau i per tant no es pot seguir mostrant la interfície gràfica del nostre programa. No es pot controlar.
- La base de dades funciona correctament i retorna els resultats de la consulta. La aplicació web pot mostrar el resultat gràficament. Funcionament normal de la aplicació.

6.2 Resultats

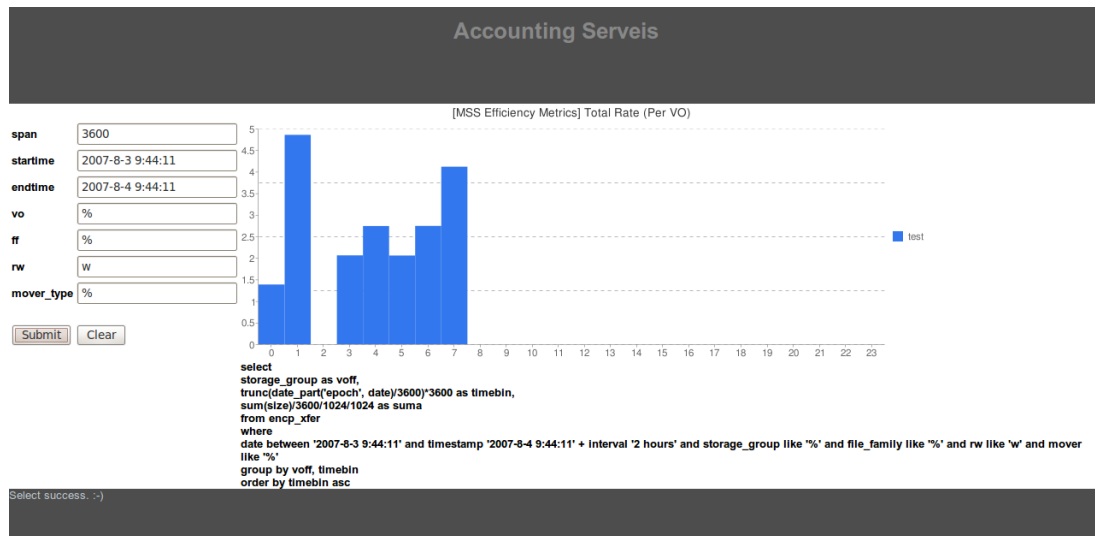


Figura 6.1: Pàgina web amb resultats de una mètrica amb poques dades

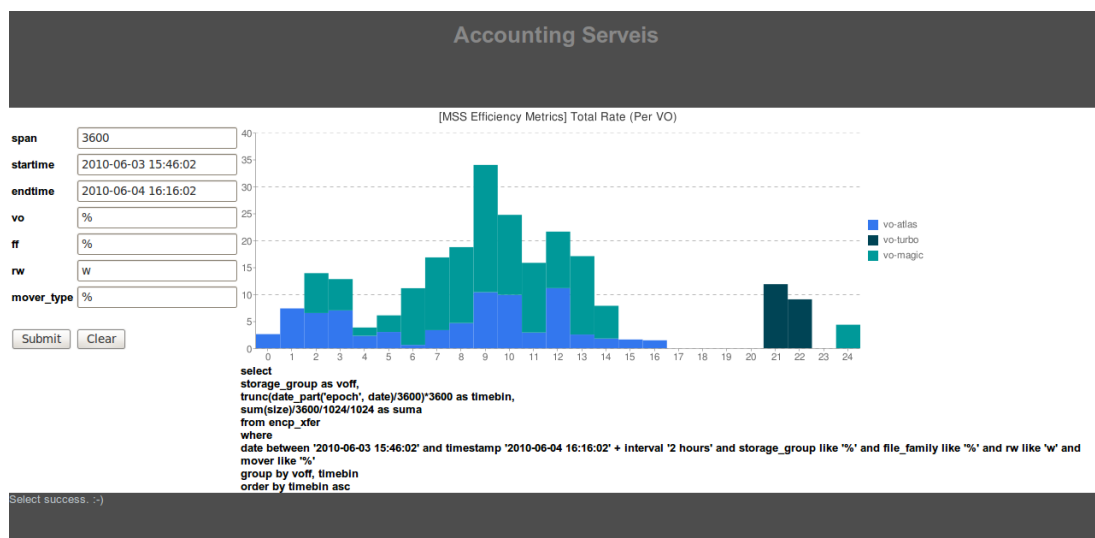


Figura 6.2: Pàgina web amb resultats de una mètrica amb moltes dades

Capítol 7

Conclusions i treball futur

La experiència de dur a terme el Projecte Final de Carrera en una institució com és el PIC, ha resultat molt interessant no només per a la proximitat amb la universitat i tots els avantatges que això proporciona, sinó també per al tipus de projecte de que es tracta. Un projecte plantejat per part de una empresa resulta més atractiu per la experiència professional que permet acreditar i el prestigi de la institució en la que s'ha realitzat. També proporciona una presa de contacte amb el camp de la Informàtica en el món professional.

Treballar en un lloc així te molts avantatges, començant per el fet d'estar treballant amb Enginyers Informàtics qualificats i experimentats i poder beneficiar-se dels seus consells en diferents camps de la Informàtica que s'han contemplat al llarg del Projecte. A més de la oportunitat de aprendre una no menyspreable varietat de tecnologies utilitzades en entorns de treball reals de investigació científica o la gestió de grans volums d'informació.

En el transcurs del projecte es va fer evident que els objectius inicials sobrepassen les possibilitats reals del temps i mitjans de que es disposava, però tot i així, els objectius primaris han estat acomplerts i s'ha creat una base des de la que es possible proseguir en un futur.

El fet de no haver assolir la totalitat dels objectius del projecte és possiblement degut a l poc coneixement de les tecnologies amb les que s'ha treballat, gairebé tot el

que s'ha utilitzat en el Projecte era nou i ha requerit un estudi previ. Exceptuant la programació en JAVA, la resta de tecnologies eren poc conegudes, la gestió de una base de dades amb PostgreSQL, la configuració d'aquesta mateixa per a garantir accés a usuaris, el desenvolupament d'aplicacions web en l'entorn de treball GWT i la configuració de un servidor web per atendre peticions de clients i interactuar amb bases de dades per transferir informació de un extrem a l'altre, etc.

També s'han hagut d'assolir uns coneixements en l'administració de sistemes Linux que no s'havien previst inicialment, en el PIC es prefereix treballar amb Linux i només es comptava amb uns coneixements bàsics de la administració d'aquests sistemes. Al mateix temps la programació en Bourne Shell ha acabat resultant atractiva i molt més versàtil del que en un començament semblava, un cop dominades les comandes ha resultat possible el desenvolupament de programes complexos en unes poques línies.

El estat del projecte a dia d'avui és que s'ha implementat i provat en entorn de test i amb dades de producció un dels sis col·lectors i la resta de components del Sistema d'accounting, la base de dades en PostgreSQL, la interfície web d'usuari, els servidor d'aplicacions amb comunicacions entre client-servidor de la base de dades a la web i la aplicació de interpretació de les dades en forma de gràfic.

D'aquesta manera, la base del projecte queda establerta i la resta de col·lectors es contempen com a futures ampliacions així com la inclusió de les seves mètriques corresponents. També la incorporació de noves mètriques per a les dades que ja es troben actualment a la base de dades i la millora de la interfície gràfica de la aplicació web.

La experiència ha estat molt gratificant, malgrat no haver aconseguit tots els objectius inicials ja sigui per falta de coneixement previ en els àmbits que s'han tractat o per uns requeriments inicials massa ambiciosos. Però es valora molt positivament el fet de realitzar el Projecte de Fi de Carrera a una empresa ja sigui en el camp de la investigació com és el PIC o en un sector més lucratiu.

Bibliografia

- [1] PIC, *Port d'Informació Científica*,
<http://www.pic.es>,
Juny 2010.
- [2] Worldwide LHC Computing Grid Technical Site, *Worldwide LHC Computing Grid*,
<http://lcg.web.cern.ch/lcg/>,
Juny 2010.
- [3] dCache project web page, *dCache project*,
<http://www.dcache.org>,
Juny 2010.
- [4] Enstore, *Fermilab Mass Storage System*,
<http://www-ccf.fnal.gov/enstore/>,
Juny 2010.
- [5] Eclipse, *Eclipse Foundation*,
<http://www.eclipse.org/org/>,
Febrer 2010.
- [6] PostgreSQL web page, *PostgreSQL Global Development Group*,
<http://www.postgresql.org>,
Juny 2010.
- [7] Google Chart API, *Google*,
http://code.google.com/intl/es-ES/apis/chart/image_charts.html,
Maig 2010.

- [8] Unix Man Pages, *University of Alabama*,
<http://bama.ua.edu/cgi-bin/man-cgi>,
Maig 2010.
- [9] PostgreSQL RPMs, *Command Prompt, Inc.*,
<https://public.commandprompt.com/projects/pgcore/wiki>,
Juny 2010.
- [10] Java Platform, Standard Edition 6 API, *Sun Microsystems*,
<http://java.sun.com/javase/6/docs/api/>,
Juny 2010.
- [11] PATH and CLASSPATH (The Java Tutorials, Essential Classes, The Platform Environment, *Sun Microsystems*,
<http://java.sun.com/docs/books/tutorial/essential/environment/paths.html>,
Abril 2010.
- [12] PostgreSQL JDBC Driver, *The PostgreSQL Global Development Group*,
<http://jdbc.postgresql.org/index.html>,
Juny 2010.
- [13] PostgreSQL warm standby, *Theo Schlossnagle*,
<http://lethargy.org/~jesus/writes/postgresql-warm-standby-on-zfs-crack>,
Juny 2010.
- [14] pgpool-II page, *PgPool Global Development Group*,
<http://pgpool.projects.postgresql.org/>,
Juny 2010.
- [15] Replication, Clustering, and Connection Pooling, *PostgreSQL Web Team*,
http://wiki.postgresql.org/wiki/Replication,_Clustering,_and_Connection_Pooling#Replication ,
Juny 2010.

- [16] SSH User Identities, *Symantec*,
<http://www.symantec.com/connect/es/articles/ssh-user-identities> ,
Juny 2010.
- [17] The Java Tutorials, PATH and CLASSPATH, *Sun Developer Network*,
<http://java.sun.com/docs/books/tutorial/essential/environment/paths.html>,
Juny 2010.
- [18] Google Web Toolkit 2.0 API, *Google*,
<http://google-web-toolkit.googlecode.com/svn/javadoc/2.0/index.html>,
Juny 2010.
- [19] Google Visualization API, *Google*,
[http://code.google.com/intl/ca-ES/apis/visualization/documentation/
using_overview.html](http://code.google.com/intl/ca-ES/apis/visualization/documentation/using_overview.html) ,
Juny 2010.
- [20] Google Code, *Google*,
<http://code.google.com/intl/ca-ES/> ,
Juny 2010.

Índex de figures

2.1	LHC Computing Grid	11
2.2	Organigrama del PIC	13
3.1	Planificació inicial (Part 1)	27
3.2	Planificació inicial (Part 2)	28
4.1	Diagrama de fluxe de un Col·lector	33
4.2	Comunicacions entre els diferents mòduls	37
5.1	Diagrama de blocs del sistema de monitoratge	40
5.2	Interfície JDBC	44
5.3	Comunicacions entre la Base de Dades d'accounting, el servidor d'aplicacions i la interfície web	45
5.4	Google Chart Tools Live Chart Playground	46
6.1	Pàgina web amb resultats de una mètrica amb poques dades	53
6.2	Pàgina web amb resultats de una mètrica amb moltes dades	53

Capítol 8

Annex 1

Casos d'ús

Cas d'us 1	Obtenir dades de dCache	
Descripció	El col·lector de dCache en el moment en ser llençat per la rutina que llenci els col·lectors haurà d'accedir a la base de dades PostgreSQL per recopilar la informació d'accounting.	
Precondició	La base de dades de accounting de dCache ha d'estar operativa.	
Seqüència Normal	Pas	Acció
	1	Conectar amb la base de dades de dCache
	2	Obtenir dades d'accounting de dCache
	3	Conectar amb la base de dades d'accounting central
	4	Desar dades obtingudes a la base de dades
	5	Tancar connexions amb les bases de dades
Postcondició	ninguna	
Excepcions	Pas	Acció
	1	En el cas que no es pugui connectar amb la BBDD de dCache el sistema haurà de reintentar-ho.
	2	En el cas que no s'obtinguin les dades d'accounting de dCache de la BBDD el sistema haurà de tancar la connexió i repetir el cas d'us #1. (Aquest)
	3	En el cas que no es pugui connectar amb la BBDD central el sistema haurà de reintentar-ho un màxim de 5 vegades abans de tancar la connexió amb dCache i reintentar el cas d'us #1.
	4	En cas que no es puguin desar les dades de dCache a la base de dades central, el sistema haurà de reintentar-ho un màxim de 5 cops abans de tancar la connexió amb la BBDD central i anar al pas #3 del cas d'us #1.
Comentaris	ningun	

Cas d'us 2	Obtenir dades de Enstore	
Descripció	El col·lector de Enstore en el moment en ser llençat per la rutina que llenca els col·lectors haurà d'accedir a la base de dades PostgreSQL per recopilar la informació d'accounting.	
Precondició	La base de dades de accounting de Enstore ha d'estar operativa.	
Seqüència Normal	Pas	Acció
	1	Conectar amb la base de dades de Enstore
	2	Obtenir dades d'accounting de Enstore
	3	Conectar amb la base de dades d'accounting central
	4	Desar dades obtingudes a la base de dades
	5	Tancar connexions amb les bases de dades
Postcondició	ninguna	
Excepcions	Pas	Acció
	1	En el cas que no es pugui connectar amb la BBDD de Enstore el sistema haurà de reintentar-ho.
	2	En el cas que no s'obtinguin les dades d'accounting de Enstore de la BBDD el sistema haurà de tancar la connexió i repetir el cas d'us #2. (Aquest)
	3	En el cas que no es pugui connectar amb la BBDD central el sistema haurà de reintentar-ho un màxim de 5 vegades abans de tancar la connexió amb Enstore i reintentar el cas d'us #2.
	4	En cas que no es puguin desar les dades de Enstore a la base de dades central, el sistema haurà de reintentar-ho un màxim de 5 cops abans de tancar la connexió amb la BBDD central i anar al pas #3 del cas d'us #2.
Comentaris	ningun	

Cas d'us 3	Obtenir dades de LFC	
Descripció	El col·lector de LFC en el moment en ser llençat per la rutina que llençi els col·lectors haurà d'accedir a la base de dades Oracle per recopilar la informació d'accounting.	
Precondició	La base de dades de accounting de LFC ha d'estar operativa.	
Seqüència Normal	Pas	Acció
	1	Conectar amb la base de dades de LFC
	2	Obtenir dades d'accounting de LFC
	3	Conectar amb la base de dades d'accounting central
	4	Desar dades obtingudes a la base de dades
	5	Tancar connexions amb les bases de dades
Postcondició	ninguna	
Excepcions	Pas	Acció
	1	En el cas que no es pugui connectar amb la BBDD de LFC el sistema haurà de reintentar-ho.
	2	En el cas que no s'obtinguin les dades d'accounting de LFC de la BBDD el sistema haurà de tancar la connexió i repetir el cas d'us #3. (Aquest)
	3	En el cas que no es pugui connectar amb la BBDD central el sistema haurà de reintentar-ho un màxim de 5 vegades abans de tancar la connexió amb LFC i reintentar el cas d'us #3.
	4	En cas que no es puguin desar les dades de LFC a la base de dades central, el sistema haurà de reintentar-ho un màxim de 5 cops abans de tancar la connexió amb la BBDD central i anar al pas #3 del cas d'us #3.
Comentaris	ningun	

Cas d'us 4	Obtenir dades de FTS	
Descripció	El col·lector de FTS en el moment en ser llençat per la rutina que llenci els col·lectors haurà d'accedir a la base de dades Oracle per recopilar la informació d'accounting.	
Precondició	La base de dades de accounting de FTS ha d'estar operativa.	
Seqüència Normal	Pas	Acció
	1	Conectar amb la base de dades de FTS
	2	Obtenir dades d'accounting de FTS
	3	Conectar amb la base de dades d'accounting central
	4	Desar dades obtingudes a la base de dades
	5	Tancar connexions amb les bases de dades
Postcondició	ninguna	
Excepcions	Pas	Acció
	1	En el cas que no es pugui connectar amb la BBDD de FTS el sistema haurà de reintentar-ho.
	2	En el cas que no s'obtinguin les dades d'accounting de FTS de la BBDD el sistema haurà de tancar la connexió i repetir el cas d'us #4. (Aquest)
	3	En el cas que no es pugui connectar amb la BBDD central el sistema haurà de reintentar-ho un màxim de 5 vegades abans de tancar la connexió amb FTS i reintentar el cas d'us #4.
	4	En cas que no es puguin desar les dades de FTS a la base de dades central, el sistema haurà de reintentar-ho un màxim de 5 cops abans de tancar la connexió amb la BBDD central i anar al pas #3 del cas d'us #4.
Comentaris	ningun	

Cas d'us 5	Obtenir dades de 3d(Oracle)	
Descripció	El col·lector de 3d(Oracle) en el moment en ser llençat per la rutina que llenca els col·lectors haurà d'accedir a la base de dades Oracle per recopilar la informació d'accounting.	
Precondició	La base de dades de accounting de 3d(Oracle) ha d'estar operativa.	
Seqüència Normal	Pas	Acció
	1	Conectar amb la base de dades de 3d(Oracle)
	2	Obtenir dades d'accounting de 3d(Oracle)
	3	Conectar amb la base de dades d'accounting central
	4	Desar dades obtingudes a la base de dades
	5	Tancar connexions amb les bases de dades
Postcondició	ninguna	
Excepcions	Pas	Acció
	1	En el cas que no es pugui connectar amb la BBDD de 3d(Oracle) el sistema haurà de reintentar-ho.
	2	En el cas que no s'obtinguin les dades d'accounting de 3d(Oracle) de la BBDD el sistema haurà de tancar la connexió i repetir el cas d'us #5. (Aquest)
	3	En el cas que no es pugui connectar amb la BBDD central el sistema haurà de reintentar-ho un màxim de 5 vegades abans de tancar la connexió amb 3d(Oracle) i reintentar el cas d'us #5.
	4	En cas que no es puguin desar les dades de 3d(Oracle) a la base de dades central, el sistema haurà de reintentar-ho un màxim de 5 cops abans de tancar la connexió amb la BBDD central i anar al pas #3 del cas d'us #5.
Comentaris	ningun	

Cas d'us 6	Obtenir dades de PBS	
Descripció	El col·lector de PBS en el moment en ser llençat per la rutina que llenci els col·lectors haurà d'accedir màquina on es desen els fitxers de log del servei, i recolectar dades dels fitxers que no s'hagin consultat ja en ocasions anteriors.	
Precondició	La màquina d'accounting de PBS ha d'estar operativa.	
Seqüència Normal	Pas	Acció
	1	Conectar amb la màquina de PBS
	2	Obtenir dades dels fitxers de log de PBS
	3	Conectar amb la base de dades d'accounting central
	4	Desar dades obtingudes a la base de dades
	5	Tancar connexió amb la base de dades d'accounting central.
Postcondició	ninguna	
Excepcions	Pas	Acció
	1	En el cas que no es pugui connectar amb la màquina de PBS el sistema haurà de reintentar-ho.
	2	En el cas que no s'obtinguin les dades d'accounting de PBS el sistema haurà de tancar la connexió i repetir el cas d'us #6. (Aquest)
	3	En el cas que no es pugui connectar amb la BBDD central el sistema haurà de reintentar-ho un màxim de 5 vegades abans de tancar la connexió amb PBS i reintentar el cas d'us #6.
	4	En cas que no es puguin desar les dades de PBS a la base de dades central, el sistema haurà de reintentar-ho un màxim de 5 cops abans de tancar la connexió amb la BBDD central i anar al pas #3 del cas d'us #6.
Comentaris	ningun	

Cas d'us 7	Llençar Col·lectors	
Descripció	El sistema permetrà que el scrip llençador dels col·lectors es dispari de forma regular i activi els col·lectors de cada un dels serveis perquè facin la seva funció específica cada un d'ells.	
Precondició	ninguna	
Seqüència Normal	Pas	Acció
	1	Fer una crida als col·lectors perquè s'executin.
Postcondició	ninguna	
Excepcions	Pas	Acció
	1	En el cas de que algun dels col·lectors no es pugui llençar, el sistema haurà de llençar un missatge d'error dient quin és el col·lector que falla i el error que dona.
Comentaris	ninguna	

Cas d'us 8	Solicitar dades	
Descripció	El sistema haurà de permetre a l'usuari final de l'aplicació que sol·liciti la visualització gràfica de una de les mètriques de la pàgina web a la qual es trobaran llistades.	
Precondició	La pàgina web haurà d'estar disponible, el servidor d'aplicacions web ha d'estar operatiu perquè això passi i ha de tenir accés a la base de dades central de accounting.	
Seqüència Normal	Pas	Acció
	1	L'usuari de la pàgina web sol·licita la visualització de una de les mètriques disponibles.
	2	El sistema sol·licita al servidor web la informació referent a la mètrica que s'ha de visualitzar.
	3	El servidor web llançara el cas d'ús #9 que consisteix en obtenir les dades de la BBDD central d'accounting
	4	Un cop es tenen les dades de la base de dades central d'accounting el sistema llença el cas d'ús #10 que consisteix en mostrar el gràfic de la mètrica sol·licitada.
Postcondició	Ninguna	
Excepcions	Pas	Acció
	2	En cas que el servidor web no doni resposta, es mostrarà un missatge d'error per pantalla a l'usuari de la web explicant l'error concret.
Comentaris	ningun	

Cas d'us 9	Consultar dades a la BBDD	
Descripció	La aplicació web consultara a la BBDD central d'accounting les dades referents a la mètrica que ha sol·licitat el usuari de la web en el cas d'us #8 i les retornara a la aplicació web en qüestió.	
Precondició	El servidor d'aplicacions web ha d'estar operatiu perque això passi i ha de tenir acces a la base de dades central de accounting.	
Seqüència Normal	Pas	Acció
	1	El servidor web rep una petició de dades per part de una de les seves aplicacions web.
	2	El servidor web fa la consulta pertinent a la base de dades per tal d'obtenir les dades de la mètrica que s'ha sol·licitat.
	3	Un cop obtingudes les dades, el servidor web les retorna perque es pugui realitzar el cas d'us #10 consistent en mostrar el resultat de la mètrica gràficament a la pàgina web.
	4	Un cop atesa la petició, el servidor web continuarà escoltant peticions.
Postcondició	Ninguna	
Excepcions	Pas	Acció
	1	Si es rep una petició que no es pot dur a terme perque no ha estat correctament tramesa, es retornarà un missatge d'error.
	2	Si la BBDD no es troba disponible, el servidor web reintentarà la consulta un màxim de 5 cops, transcorreguts els quals retornarà un missatge d'error conforme no es te acces a la BBDD.
Comentaris	ningun	

Cas d'us 10	Mostrar Gràfic	
Descripció	La aplicació web mostrarà els resultats de la consulta pertinent a la mètrica que hagi escollit el usuari de la web. Aquest resultats gàfics es generaran des del servidor d'aplicacions web mitjançant l'ús del GWT.	
Precondició	El servidor d'aplicacions web ha d'estar operatiu perquè això passi i ha de tenir accés a la base de dades central de accounting. La Base de dades ha d'haver retornat els resultats de la consulta pertinent per poder generar la gràfica de la mètrica sol·licitada.	
Seqüència Normal	Pas	Acció
	1	La aplicació web rebrà la informació de la base de dades que haurà sol·licitat en el cas d'us #8
	2	Amb les dades rebudes generarà una gràfica del tipus sol·licitat per el usuari de l'aplicacio
Postcondició	Ninguna	
Excepcions	Pas	Acció
	1	Si la informació rebuda no concorda amb el format sol·licitat, es retornarà un missatge d'error.
Comentaris	ningun	

Signat:

Ignasi Sans Claret

El Port d'Informació Científica és un centre de Computació Grid de referència que dona suport a comunitats científiques, com el LHC (CERN). Al PIC, trobem una gran varietat de tecnologies que proporcionen serveis al centre. Des de la arquitectura i elements de la xarxa, fins a recursos informàtics de computació, sistemes d'emmagatzematge a disc i cinta magnètica, bases de dades (ORACLE/PostgreSQL). El projecte consisteix en el disseny i implementació d'una base de dades col·lectora de tota la informació rellevant dels diferents sistemes del centre, i un portal web on mostrar tots els valors i gràfiques, tot basat en programari lliure.

El Port d'Informació Científica es un centro de Computación Grid de referencia que da soporte a comunidades científicas, como el LHC (CERN). En el PIC, encontramos una gran variedad de tecnologías que proporcionan servicios en el centro. Desde la arquitectura y elementos de la red, hasta recursos informáticos de computación, sistemas de almacenamiento en disco y cinta magnética, bases de datos (ORACLE / PostgreSQL). El proyecto consiste en el diseño e implementación de una base de datos colectora de toda la información relevante de los diferentes sistemas del centro, y un portal web donde mostrar todos los valores y gráficos, todo basado en software libre.

The Port d'Informació Científica is a reference Computing Grid center that supports scientific communities, such as the LHC (CERN). A wide variety of technologies that provide services to the center. From architecture and network elements, to computing resources, disk and tape storage systems, databases (Oracle / PostgreSQL). The project involves the design and implementation of a database collecting all the relevant information of different systems from the center and a website to show all values and graphs, all based on free software.