

Collecting and disseminating CDS KPIs

Introduction

CDS (CERN Document Server) stores over 900,000 bibliographic records, including 360,000 fulltext documents, of interest to people working in particle physics and related areas. My project consisted on extracting KPIs (Key Performance Indicators) from it and feeding them to a central IT KPI system. To achieve this, I learned the CDS-Invenio open source digital library software, and worked with its statistic module.

Description of the project

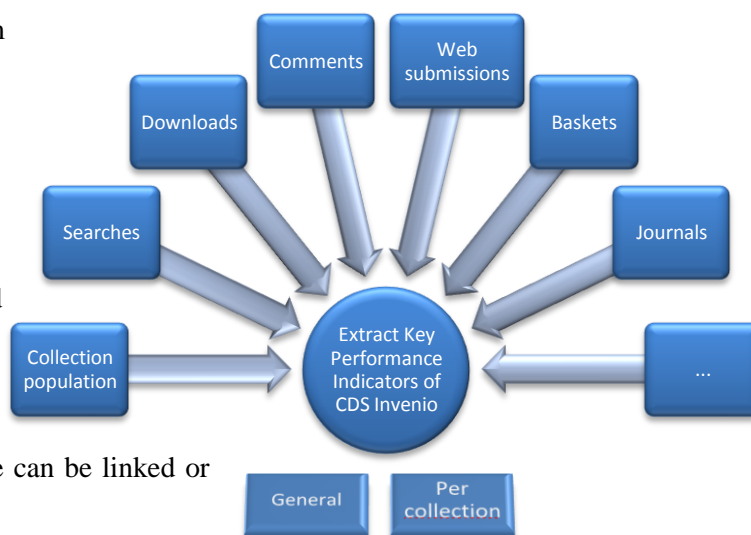
This project was divided in three main parts:

1. Extract KPIs of CDS Invenio
2. Develop reports for the librarians
3. Export to central CERN monitoring system

Extract KPIs of CDS Invenio

My first task was to get the main numbers and statistics from CDS. These included collection population, searches, downloads, comments, web submissions, baskets, views of articles in journals...

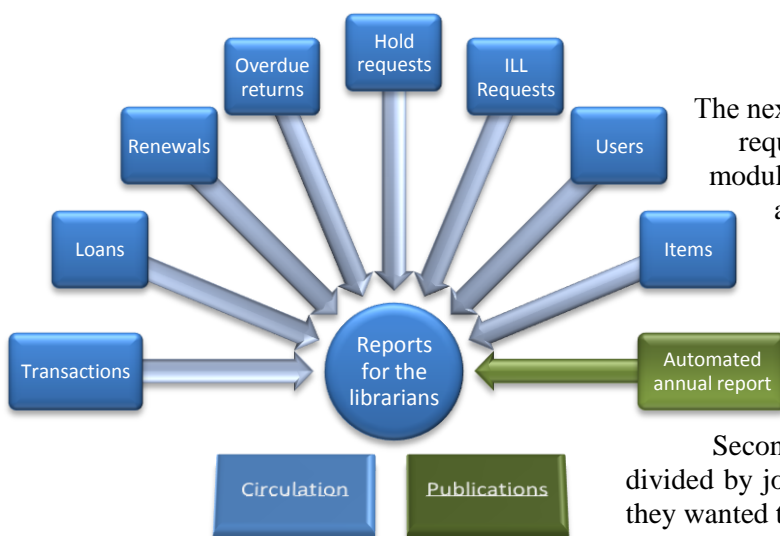
I had to do this both for the whole system and specific per collection. At this point, I also developed a standard statistics page per collection (with the number of documents, downloads and comments in the collection during the last month). This default statistics page can be linked or shown in some way in each collection's webpage.



Develop reports for the librarians

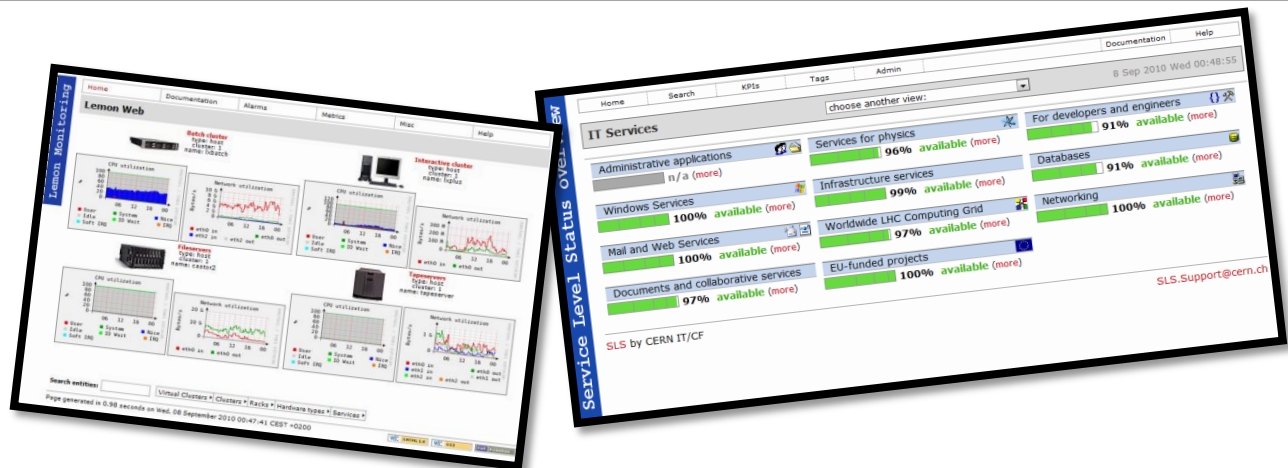
The next step in this project was a consequence of the librarians' requests. Firstly, they wanted statistics on the BibCirculation module of CDS Invenio, which manages the loans. I developed annual and monthly reports on the transactions (loans and ILL requests), statistics about loans (number of loans, most loaned documents, documents never loaned...), renewals, overdue returns, hold requests (requests for a record which is on loan), ILL requests (between libraries), active users and items (physical representation of a record).

Secondly, there is an annual report on CERN publications divided by journals that the librarians create manually until now and they wanted to generate it automatically.



Export to central CERN monitoring system

The last step in my project involved exporting the statistics collected from CDS Invenio to one of the CERN monitoring systems. The two main options we looked into were [Lemon Monitoring System](#) and [Service Level Status](#).



Understanding WebStat

CDS Invenio consists of several more or less independent modules with precisely defined functionality. Within these modules, WebStat is a configurable system that permits to gather statistics about the health of the server, the usage of the system, as well as about some particular system features. This is the module I mainly worked on. WebStat provides two main kinds of statistics and many output formats.

Key statistics

These are numbers extracted from the existing data. Nowadays, we mainly use two sources:

- The database tables: number of documents, number and type of searches...
- The records (MARC data): loans filtered by date of publication (field 260\$\$c)...

Custom events

These statistics register new actions. They are much more customizable than the key statistics, because we can record new data, so we can choose any element we want to have statistics of.

I will now show an example of how to add a custom event. In this case, we want to see the number of views of the articles in a journal. The first step consists of modifying the webstat.cfg configuration file, adding the following entry:

```
[webstat_custom_event_1]
name = journals
param1 = action
param2 = bulletin
param3 = issue_number
param4 = category
param5 = language
param6 = articleid
```

After that, we have to execute `inveniocfg --load-webstat-conf` so the necessary tables are created. Then, we have to plug one of the WebStat functions (`register_customevent`) where the article is displayed in the WebJournal module. The line of code we should add looks like this:

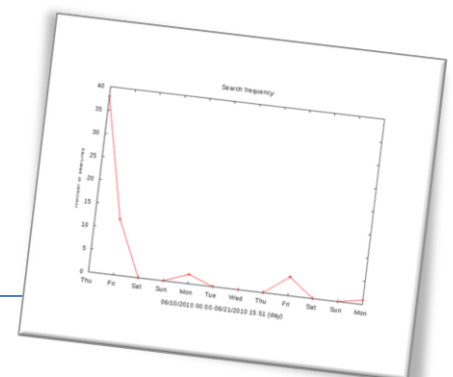
```
register_customevent("journals", ["display", "AtlantisTimes", "02/2009", "News", "en", "101"])
```

Output formats

Graphs

Gnuplot

This kind of graphs was available when I started improving the WebStat module. It generates an image file that is then displayed in the web page.



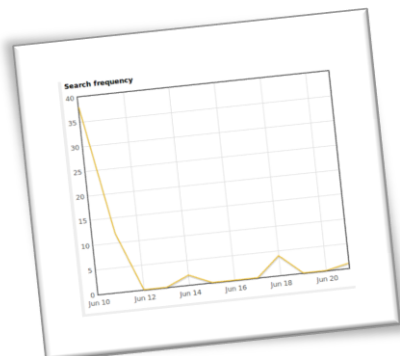
Collecting and disseminating CDS KPIs

Carmen Álvarez Pérez

carmen.alvarez.perez@cern.ch



IT-UDS-CDS

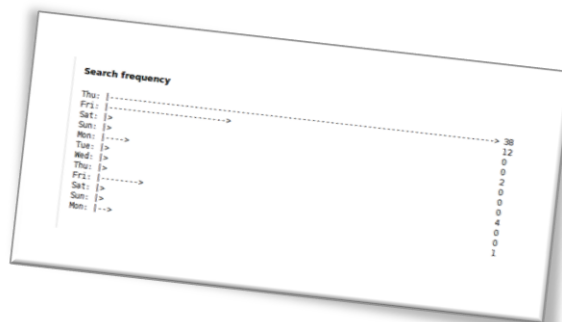


Flot

This is a Javascript plotting library. I added this kind of graphs because it is much more attractive, and it is also interactive (allows zooming and tooltips). Most of the people will be able to use this one, but as it is Javascript based, some may have problems. That is why the other graph formats are still available. I also implemented this kind of graphs in other parts of CDS Invenio (the BibRank module), where only Gnuplot was available.

ASCII art

This is a text-based graph. It was already available before I started working on the WebStat module.



Tables and lists

Loans statistics	
Number of documents loaned	10
Number of items loaned on the total number of items	0.666666666667
Number of items never loaned on the total number of items	0.333333333333
Average time between the date of the record creation and the date of the first loan	57

Tables

The librarians requested this kind of output for the BibCirculation statistics.

Lists

The librarians also requested lists for some of their statistics.

Users lists			
Name	Address	Mailbox	E-mail
Balthasar Montague	20-M-349	None	balthasar.montague@cds.cern.ch
Benvolio Montague	93-P-019	None	benvolio.montague@cds.cern.ch
Romeo Montague	98-W-859	None	romeo.montague@cds.cern.ch
Dorian Gray	38-Y-819	None	dorian.gray@cds.cern.ch

Data

There were two export formats when I started with this project: CSV (Comma Separated Values) and Python code. As a request from the librarians, I added Excel export for the statistics that allowed lists output.

Exporting KPIs



SLS

- Simple
- XML
- Services

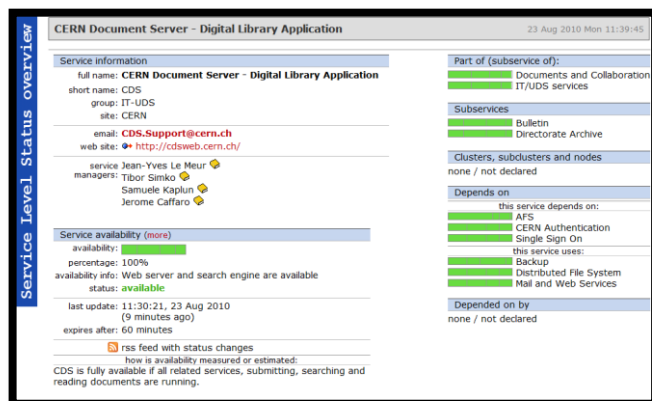
We finally chose to use SLS as the system to export our KPIs. The decision was based on the difficulty of adding the data to the system, the language I had to use and the main orientation of the monitoring system. SLS turned out to be much simpler, the language to use is XML and it is service oriented.

Lemon

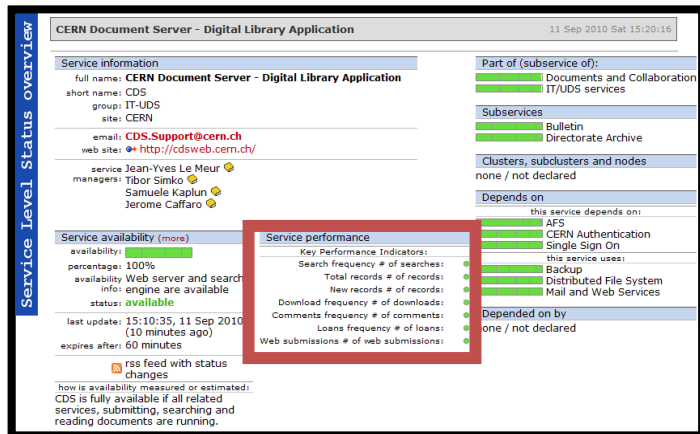
- Complex
- Perl
- Hardware



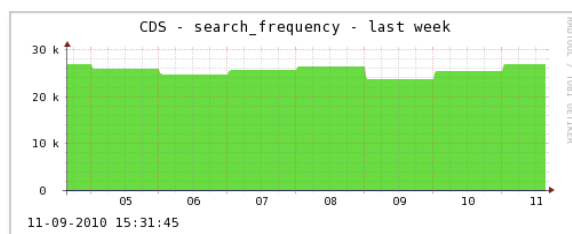
CDS already had a [page](#) in SLS. It showed information about the service, the availability, dependencies...



Now, it also shows the service performance, which includes the KPIs extracted from CDS.



If you click on one of the KPIs, you can see a graph like this:



To add this data to SLS, I developed a script which generates an XML like this one (I added more KPIs):

```
<?xml version="1.0" encoding="UTF-8" ?>
<serviceaccountinginfo
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://sls.cern.ch/SLS/XML/accounting
    http://sls.cern.ch/SLS/XML/accounting.xsd"
  xmlns="http://sls.cern.ch/SLS/XML/accounting">
  <id>CDS</id>
  <day>2010-08-15</day>
  <kpis>
    <kpi id="search_frequency" name="Search frequency"
      type="numeric" unit="# of searches" goal="more">
      <target>0</target>
      <value>5214</value>
    </kpi>
  </kpis>
</serviceaccountinginfo>
```

This is how it looks

Custom event

I will keep on using the views of the articles on a journal as example. In this case, we can see how it is displayed using the Flot output format. We can also specify different parameters, as the time span and values for the different arguments. We can even display more than one custom event in one graph. The small graph is used for zooming. I will not show here the key statistics display as it is similar to this one.



Collecting and disseminating CDS KPIs

Carmen Álvarez Pérez

carmen.alvarez.perez@cern.ch



IT-UDS-CDS

Tables and lists

We can see now examples of tables and lists:

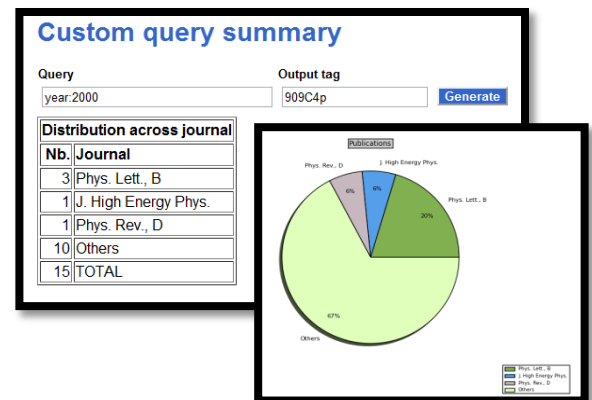
Loans statistics					
User address	UDC	Creation date	Item status	Publication date	Time span
- select User address	- select UDC	- select Creation date	On loan	- select Publication date	This month
Generate					
Loans statistics					
Number of documents loaned			7		
Number of items loaned on the total number of items			0.388888888889		
Number of items never loaned on the total number of items			0.611111111111		
Average time between the date of the record creation and the date of the first loan			57		

Loans request lists			
User address	UDC	Time span	
- select User address	- select UDC	Select date...	
		From:	07/01/2010 00:00
		To:	08/15/2010 14:00
Generate Excel			
Hold requests lists			
Title	Author	Edition	Barcode
Introductory statistics a decision map	Harshbarger, Thad R	2nd ed	bc-27001
Introduction to metamathematics	Kleene, Stephen Cole		bc-29001
Klystrons and microwave triodes	Hamilton, Donald R		bc-22001
Analyse informatique t.2 L'accomplissement	Dasse, Michel		bc-26001

Custom query summary

Finally, I want to show an example of the custom query summary, which I created from the librarians' request for an automated annual report of CERN publications on journals. The result was a table, in which you can specify a query for the results that will be shown and an output tag for the distribution of them.

In the example, we use the query `year:2000` and the output tag `909C4p`, which is the MARC tag for journal name. The result is a table of all the records from 2000 distributed across journals. A pie chart is also generated.



Side project

I was also assigned the project of improving portal boxes. Portal boxes are text boxes that administrators can add and customize for each collection webpage. For example, the text in the red frame is a portal box. The two main needs were a better language support and the possibility to add python code in them. I managed to complete both improvements.

Conclusions

As the main conclusion, the Invenio WebStat module has been improved. Now there are more output formats, key statistics and export options available. Also, the main statistics of CDS Invenio are now out of the box (collection population, searches, downloads, comments, web submissions, baskets, journals, circulation and publications). The KPIs of CDS are now available in SLS, and the rest of the project is being committed and integrated for its future deploy in CDS.

As a personal evaluation, I have improved my skills to work in a team and to speak in front of an audience. I also had the opportunity to see how a real big software project has been designed (CDS Invenio), and to dig into some of its modules to learn how they work. Finally, I learned the Python programming language, which I had not use before, and renewed my knowledge of Javascript and SQL.

Future work

Possible improvements on the WebStat module include any request from other modules (as for circulation and publications), or other numbers that can be extracted with web log analyzers like AWStats.

