Metrics

Metrics infrastructure

This feature has been introduced in DSpace-CRIS 5.5

The system allows to store via JAVA API metrics about any object in the table cris_metrics

```
CREATE TABLE public.cris_metrics
(
  id integer NOT NULL,
  metriccount double precision NOT NULL,
  enddate timestamp without time zone,
  remark text,
  resourceid integer,
  resourcetypeid integer,
  startdate timestamp without time zone,
  timestampcreated timestamp without time zone,
  timestamplastmodified timestamp without time zone,
  metrictype character varying(255),
  uuid character varying(255),
  last boolean,
  CONSTRAINT cris_metrics_pkey PRIMARY KEY (id)
)
```

metriccount holds the metric value such as the number of received citation, downloads, etc.

metrictype holds an alias of the indicator that is recorded, like view, scopus, etc.

startdate / enddate are used to store the observed period to evaluate the metric (for example the number of view from 1/1/2016 to 31/12/2016)

timestampcreated / timestamplastmodified holds the timestamp of the record creation and update

resourceid, resourcetypeid, uuid all refers to the object to whom the metric belong

last is a flag to mark the latest available value for a specific metrictype and object

remark is a text field that holds custom additional information in JSON format specific of the indicator such as the target URL to use as link for the metric value (i.e. the usage statistics URL for usage statistics metrics, the URL of the Scopus preview page for the scopus citation count, etc.). The remark field can holds anything that could be useful to the view to provide a more sophisticated and specific visualization on an indicator basis

The last metric value are available as fields in the SOLR search through the use of a dynamicfield **crismetrics_*** (i.e. crismetrics_view, crismetrics_scopus, etc.) that can be retrieved as extraInfo when the SOLR core is queried or used to sort the results

in the **dspace-installDir/solr/search/conf/schema.xml** we have added

```
<types>
        ...
        <fieldType name="crismetrics_t" class="org.dspace.solr.schema.CrisMetricsFieldType" coreName="${solr.
core.name}" />
</types>

<fields>
        ...
        <dynamicField name="crismetrics_*" type="crismetrics_t" indexed="true" stored="true"/>
</fields>
```

The ***crismetrics_**** fields in the SOLR search core are populated using a custom SOLR listner *org.dspace.solr.util.CrisMetricsUpdateListener* and component *org.dspace.solr.handler.component.CrisMetricsExtractComponent*. It performs a query on the database to retrieve all the current metrics values putting them in a SOLR cache anytime that a new Searcher is created. This mean that

- no DB queries are executed when you retrieve the metrics value from SOLR
- the field is not a "normal" SOLR field so it cannot be used in any other SOLR operation than projection (READ the value) and sorting
- update to the database are not visible until the SOLR cache is forced to refresh or a new SOLR Searcher is opened and the document ID reassigned
    - to force a refresh you can perform any query against SOLR including the special field ***clearcache-crismetrics*** in the requested fieldlist (***fl*** parameter)
    - a new SOLR Searcher is typically created when SOLR write to the disk reassigning the docs ID (as in an optimize request)

This infrastructure is currently used to store views and downloads metrics about any object and citation count retrieved from external bibliometrics database such PubMed Central, Scopus and Web of Science.

The data collection are performed by batch scripts, below you will find the technical details to decide which depending on your configuration which script and which order and frequency of execution you need. The list of CRON Job for the default installation of DSpace-CRIS is available here

Generic metric builders are configured as spring beans in the **[dspace-installDir]/config/spring/cris-metrics.xml** file and can be exectued using the script

```
org.dspace.app.cris.batch.ScriptStatsMetrics
```

it is possible to limit the script to a specific metric using the option *-s* with the name assigned to the plugin

Usage statistics as metrics

- org.dspace.app.cris.statistics.plugin.StatsViewIndicatorsPlugin

Defining a spring bean that instance such class the usage statistics for the entity specified by the resourceTypeId field are "converted" in metrics with metrictype **view** and **download**

```
<bean class="org.dspace.app.cris.statistics.plugin.StatsViewIndicatorsPlugin" name="
ItemStatsViewIndicatorsPlugin">
                  <property name="name" value="ItemStatsViewIndicatorsPlugin"/>
                  <property name="resourceTypeId" value="2"/>
</bean>
```

Content statistics as metrics

This feature has been introduced in DSpace-CRIS 5.6

- org.dspace.app.cris.statistics.plugin.IndicatorCountBuilder<ACO>
- org.dspace.app.cris.statistics.plugin.IndicatorMetadataSumBuilder<ACO>

Citation counts

The following scripts are used to query the external bibliometric databases:

PubMed Central

org.dspace.app.cris.metrics.pmc.script.RetrieveCitationInPMC

Scopus

org.dspace.app.cris.metrics.scopus.script.ScriptRetrieveCitation

Web of Science

org.dspace.app.cris.metrics.wos.script.ScriptRetrieveCitation

all the scripts accept the following parameter

- **-t** <time-in-seconds> to exclude from the processing the publications that were already checked in the database by less than the value specified in seconds. -t 604800 mean 7 days
- **-x** <number-of-publication> to limit the number of processed publications to maximum n
- **-q** <solr-query> to limit the set of publications to the ones matching the SOLR query (useful for testing and precise check/refresh)

Derivative metrics

The system is able to use existent metrics to build derivative metrics aggregating the information to an upper level as for example the total number of citation received by all the researcher's publications. This is the publication citation count metric aggregated to the researcher level. Other than aggregation it is possible to calculate average, maximum, minimum, standard deviation, variation in a week and month and also sum different metrics together.

- org.dspace.app.cris.statistics.plugin.StatsAggregateIndicatorsPlugin

```
<bean class="org.dspace.app.cris.statistics.plugin.StatsAggregateIndicatorsPlugin" name="
RPStatsAggregatorPUBMEDIndicatorsPlugin">
                  <property name="name" value="RPStatsAggregatorPUBMEDIndicatorsPlugin"/>
                  <property name="type" value="pubmed"/>
                  <property name="crisEntityClazz" value="org.dspace.app.cris.model.ResearcherPage"/>
                  <property name="crisEntityTypeId" value="9"/>
</bean>
```

the above example compute the sum of all the pubmed (citation counts) metric on publication for a specific researcher. By default the author_authority solr field is used to retrieve all the items related to a researcher but the used field can be configured as property of the spring bean and a filter query defined to limit the set of publications used (for example to the ones published in the last 5 years)

- org.dspace.app.cris.statistics.plugin.StatsPercentileIndicatorsPlugin

```
<bean class="org.dspace.app.cris.statistics.plugin.StatsPercentileIndicatorsPlugin" name="
PercentileViewIndicatorsPlugin">
                  <property name="name" value="PercentileViewIndicatorsPlugin"/>
                  <property name="metrics" value="view"/>
                  <property name="fq">
                          <list>
                                  <value>search.resourcetype:2</value>
                          </list>
                  </property>
</bean>
```

the above example compute the position of the object compared to the other object for a specific metric in percentage so to be used to state the percentile of the object for the metric. For example if a publication is the 10th over 100 publication using the number of view this mean that the view percentile is 10 /100 = 0,1

- org.dspace.app.cris.statistics.plugin.StatsPeriodIndicatorsPlugin

```
          <bean class="org.dspace.app.cris.statistics.plugin.StatsPeriodIndicatorsPlugin" name="
ItemStatsPeriodWeekPUBMEDIndicatorsPlugin">
                  <property name="name" value="ItemStatsPeriodWeekPUBMEDIndicatorsPlugin"/>
                  <property name="type" value="pubmed"/>
                  <property name="frequency" value="_last1"/>
          </bean>
```

the above example compute the variation (increment or decrements) of the pubmed metric (type) over one week (frequency = _last1, use _last2 for one month)

Since DSpace-CRIS 5.6 an additional generic plugin *org.dspace.app.cris.statistics.plugin.StatsGenericIndicatorsPlugin* is available to make further computation as defined in the following Indicators:

- org.dspace.app.cris.statistics.plugin.IndicatorMetricSumBuilder<ACO>
- org.dspace.app.cris.statistics.plugin.IndicatorMetricRatioBuilder<ACO>
- org.dspace.app.cris.statistics.plugin.IndicatorMetricPercentageBuilder<ACO>
- org.dspace.app.cris.statistics.plugin.IndicatorMetricMathBuilder<ACO>