CommonCrawlDataDumper

Introduction

The *CommonCrawlDataDumper* is a Nutch tool. It is an alias for `org.apache.nutch.tools.CommonCrawlDataDumper`. By using this tool, we can dump out Nutch segments into Common Crawl data format, mapping each crawled-by-Nutch file on a JSON-based data structure. CommonCrawlDataDumper dumps out the files and serialize them with CBOR encoding, a data representation format used in many contexts. Optionally, the CommonCrawlDataDumper is able to create a compressed archive including all CBOR-encoded data using gzip.

In order to run the CommonCrawlDataDumper tool, we can use either the command line (`bin/nutch commoncrawldump`) or the Java class (`CommonCrawlDataDumper`).

For more details on CommonCrawlDataDumper development, visit NUTCH-1949 JIRA issue.

Table of Contents

Common Crawl format

The Common Crawl corpus contains petabytes of data collected over the last 7 years. It contains raw web page data, extracted metadata and text extractions.

Common Crawl currently stores the crawl data using the Web ARChive (WARC) format. The WARC format allows for more efficient storage and processing of CommonCrawl's free multi-billion page web archives, which can be hundreds of terabytes in size. More in depth, the Common Crawl format includes three file formats:

- WARC files which store the raw crawl data
- WAT files which store computed metadata for the data stored in the WARC
- WET files which store extracted plaintext from the data stored in the WARC

Common Crawl provides a corpus for collaborative research, analysis and education, giving the great opportunity to easily access high quality crawl data that was previously only available to large search engine corporations. For this reason, dumping out Nutch segments into Common Crawl format may allow the Nutch community to contribute in realizing the Common Crawl's mission. Thus, the CommonCrawlDataDumper tool aims at providing an easy way to dump out Nutch segments into Common Crawl format.

CommonCrawlDataDumper

Currently, the CommonCrawlDataDumper tool represents a preliminary solution that maps each crawled-by-Nutch file on a JSON-based data structure including data, metadata and crawling information:

```
{
    "url": "http:\/\/somepage.com\/22\/14560817",
    "timestamp": "1411623696000",
    "request": {
        "method": "GET",
        "client": {
            "hostname": "crawler01.local",
            "address": "74.347.129.200",
            "software": "Apache Nutch v1.10",
            "robots": "classic",
            "contact": {
                "name": "Nutch Admin",
                "email": "nutch.pro@nutchadmin.org"
            }
        },
        "headers": {
            "Accept": "text\/html,application\/xhtml+xml,application\/xml",
            "Accept-Encoding": "gzip,deflate,sdch",
            "Accept-Language": "en-US,en",
            "User-Agent": "Mozilla\/5.0",
            "...": "..."
        },
        "body": null
    },
    "response": {
        "status": "200",
        "server": {
            "hostname": "somepage.com",
            "address": "55.33.51.19",
        },
        "headers": {
            "Content-Encoding": "gzip",
            "Content-Type": "text\/html",
            "Date": "Thu, 25 Sep 2014 04:16:58 GMT",
            "Expires": "Thu, 25 Sep 2014 04:16:57 GMT",
            "Server": "nginx",
            "...": "..."
        },
        "body": "\r\n  <!DOCTYPE html PUBLIC ... \r\n\r\n  \r\n    </body>\r\n    </html>\r\n  \r\n\r\n",
    },
    "key": "com_somepage_33a3e36bbef59c2a5242c2ccee59239ab30d51f3_1411623696000",
    "imported": "1411623698000"
}
```

As JSON format above, the tool does not provide yet data that perfectly adheres to Common Crawl format. This preliminary version of the tool has been released to allow the Nutch community to give important feedback and ideas in order to extend this solution. For more details, visit NUTCH-1949 JIRA issue.

The CommonCrawlDataDumper tool is able to perfom the following steps:

1. deserialize the crawled data from Nutch 2. map serialized data on the proper JSON structure 3. serialize the data into CBOR format 4. optionally, compress the serialized data using gzip

In case of exporting to an actual WARC file the process is a little different:

1. deserialize the crawled data from Nutch 2. map deserialized data into a WARC file that can be compressed or not.

The following diagram describes the workflow of CommonCrawlDataDumper tool. It includes also some annotations.

CommonCrawlDataDumper_v02.png|alt=200px title=200px!

This tool is able to work with either single Nutch segments or directory including segments as input data.

As shown in following sections, we can use either the command line (`bin/nutch commoncrawldump`) or the Java class (`CommonCrawlDataDumper`).

Using the command line

We can run the CommonCrawlDataDumper tool using the command `bin/nutch commoncrawldump`. Typing `bin/nutch commoncrawldump` without any argument, we can show the following:

```
usage: org.apache.nutch.tools.CommonCrawlDataDumper [-epochFilename]
       [-extension <extension>] [-gzip] [-h] [-jsonArray] [-keyPrefix
       <keyPrefix>] [-mimetype <mimetype>] [-outputDir <outputDir>]
       [-reverseKey] [-segment <segment>] [-SimpleDateFormat] [-warc]
       [-warcSize <warcSize>]
 -epochFilename           an optional format for output filename.
 -extension <extension>   an optional file extension for output documents.
 -gzip                    an optional flag indicating whether to
                          additionally gzip the data.
 -h,--help                show this help message.

 -jsonArray               an optional format for JSON output.
 -keyPrefix <keyPrefix>   an optional prefix for key in the output format.
 -mimetype <mimetype>     an optional list of mimetypes to dump, excluding
                          all others. Defaults to all.
 -outputDir <outputDir>   output directory (which will be created) to host
                          the CBOR data.
 -reverseKey              an optional format for key value in JSON output.
 -segment <segment>       the segment or directory containing segments to
                          use
 -SimpleDateFormat        an optional format for timestamp in GMT epoch
                          milliseconds.
 -warc                    export to a WARC file
 -warcSize <warcSize>     an optional file size in bytes for the WARC
                          file(s)
```

For example, we can run the tool against Nutch segments in `/path/to/input_dir` by typing:

```
/bin/nutch commoncrawldump -outputDir /path/to/output_dir -segment /path/to/input_dir -mimetype pdf -gzip
```

The command above dumps out the PDF files, excluding all other mimetypes in Nutch segments located on `/path/to/input_dir`, creating a `.tar.gz` archive in `/path/to/output_dir`. The gzip archive contains all CBOR-encoded files extracted from input segments. Actually, only `-outputDir` and `-segment` command-line options are mandatory.

Using the CommonCrawlDataDumper Java class

CommonCrawlDataDumper.java (`org.apache.nutch.tools.CommonCrawlDataDumper.java`) is the file containing the Java implementation of the tool. In addition to the entry point (`main`), this Java class provide only one public method called `dump`:

```
public void dump(File outputDir, File segmentRootDir, boolean gzip, String[] mimeTypes, boolean warc) throws
Exception {
    // code
}
```

This method implements the core task of CommonCrawlDataDumper. It accepts five arguments: `outputDir` is the output directory to save CBOR-encoded data, `segmentRootDir` is the input directory including Nutch segments, `gzip` determines if compression is used, `mimeTypes` contains a list of mimetypes, if provided, and `warc` determines if the export must be donde into an actual WARC file instead of the CBOR based format. We can call the CommonCrawlDataDumper tool from a Java program using this method.

Example

If Nutch has been installed correctly, we can start crawling by typing the following command:

```
bin/crawl urls/ testCrawl/ http://localhost:8983/solr/ 2
```

The command above allows to create Nutch segments for crawled data in `testCrawl` folder. We can use this folder as input for CommonCrawlDataDumper in order to dump out data crawled using Nutch. Obviously, the `crawl` command is not necessary if we have already Nutch segments to dump out.

In order to dump out Nutch segments, we can use the command-line program:

```
bin/nutch commoncrawldump -outputDir outCommonCrawl -segment testCrawl/segments
```

If when you start running the script later you start getting an error called `OutOfMemoryError`, try changing the JAVA_HEAP_MAX variable in line 128 of `bin/nutch` to an appropriate value.

The `bin/nutch commoncrawldump` program dumps out all Nutch segments included in `testCrawl/segments` to `outCommonCrawl` folder, making one CBOR-encoded file for each crawled file. The tool will show a short report as follows:

```
TOTAL Stats:
{
    {"mimeType":"text/plain","count":1"}
    {"mimeType":"application/xhtml+xml","count":3"}
    {"mimeType":"application/octet-stream","count":8"}
    {"mimeType":"text/html","count":38"}
}
```

We can use also the `-gzip` and `-mimetype` options to enable compression and mimetype filtering respectively.

The `-warcSize` paramter allow to specify a maximun file size in bytes for the WARC output file, if this parameter is not specified then a default of 1GB is used as suggested in the WARC specification. If the size of the export file exceeded the `-warcSize` parameter (or the 1GB default size) then the output file will be splited in several files.

The output file is named using the following convention:

```
"${prefix}-${timestamp17}-${serialno}"
```

in this case the

```
${prefix}
```

used is `WEB` followed by the timestamp and `serialno` will be incremented each time that the file needs to be splitted to keep the size of the file below the defined limit.

If the `-gzip` option is used along with the `-warc` option the output WARC file will be compressed using the GZIP format.

Features

CBOR encoding

CBOR (RFC 7049 Concise Binary Object Representation) provides an object encoding format for serialization purposes. CBOR encoding is really simple, because it stores the information itself also in the first byte when it’s small enough. So the encoding is really comprehensive in contrast to most other encodings.

CBOR data is more compact than JSON (or other formats) when (1) numbers are used as identifiers instead of strings as JSON format, (2) complex aggregated data are used, (3) heterogeneous datatypes are used because, for example, `true` and `false` values may be represented using less bytes, and so on.

Actually, the CommonCrawlDataDumper tool wraps a single string value (corresponding to JSON-based representation of deserialized Nutch data) into CBOR.

GZip Compression

File compression allows to save much space when facing several files. To assist with large crawling tasks, the CommonCrawlDataDumper tool is able to generate (if `-gzip` option is provided) a `.tar.gz` archive including all CBOR-encoded files. The archive is named using the current timestamp (yyyyMMddhhmm.tar.gz). The tool relies on Apache Commons Compress to create `.tar.gz` archive.

WARC files

Regarding the implemented format explained above you can also export the data into a WARC file, using the `-warc` option and optionally indicate a file size using the `-warcSize` parameter. The output of this tool will be a valid WARC file with resource, request or response records.

If you need to export response/request records then you'll need to enable the `store.http.headers` and `store.http.request` settings in your `nutch-site.xml` file. This settings are **DISABLED** by default. If the raw request is not found in your segments then no request record will be generated, on the other hand if no header information is found in the segments then instead of a response record you'll get a basic resource record that will not contain any information about the response headers, but will contain the raw response from the server.

An alternative implementation of a WARC Exporter is provided in NUTCH-2102 that is shipped with Nutch (as of version 1.11). To use this tool you can use the following command to display the usage information `bin/nutch org.apache.nutch.tools.warc.WARCExporter`

```
    Usage: WARCExporter <output> (<segment> ... | -dir <segments>)
```

Essentially an output directory needs to be specified as a first parameter and a list of segments specified next, or instead use the `-dir` parameter to indicate the segments parent directory.

The WARCExporter tool is built using the Hadoop Map/Reduce approach so it could be a more appealing alternative if you're dealing with a lot of data or running nutch in a Hadoop cluster.

Future Work

CommonCrawlDataDumper is a Nutch tool under development. Currently, we provide a preliminary version to get feedback and ideas by Nutch community. Please contribute in CommonCrawlDataDumper by writing/commenting on NUTCH-1949 JIRA issue.