QOMPLX:OS

Q:Extractor

Crawl, aggregate and structure any web data

Background

The rapid growth of publicly available or visible data offers much promise but the lack of consistent structure and the continued non-emergence of semantic labelling on the web limits practical collection and use for most organizations.

With increasingly interactive web experiences like single page applications and rich server-side interaction, the ability to collect meaningful and actionable data has grown in cost, difficulty, and become less accessible. Target data is stored in varied ways from structured to unstructured, in various formats such as web pages, CSV, PDF, ppt and a myriad of other forms which are then stored in disparate locations such as on the open web, inside corporate networks, or within third-party systems.

To derive relevant insights from this vast trove of potential information, businesses must be able to extract, structure, and channel this information into their analytical frameworks while keeping track of its original source, associated restrictions on persistence or use, and the degree of confidence they may have in its authenticity or completeness. The ability to effectively extract data from widely disparate and heterogeneous sources plays a critical role within the QOMPLX Q:OS platform.

QOMPLX's Q:Extractor leverages existing, human-oriented presentation interfaces in active web pages to allow automated processes to collect and aggregate any data available to human users. It removes the barriers to data acquisition by facilitating the retrieval, refresh, and storage of structured and unstructured data not readily available from a central source. This type of data extraction is typically performed through the use of web crawlers that are configured as a set of routines intended to mimic actions of a human user browsing the web and then capturing server response. QOMPLX Q:Extractor is linearly scalable and capable of interacting with sophisticated web applications containing client-side scripting and asynchronous server communication. Its integration with the Q:OS platform ensures accessibility to all retrieved data which is readily available to provide additional value and insight to the entire platform.

Our approach to data extraction includes a rich querying mechanism, that exposes information contained within both the Document Object Model (DOM, which represents web data as a tree structure) and other web-based resources (such as .pdf or .csv files), that outputs structured data (e.g. relational or tree-based, parsed from the web-based presentation).

This query formalism includes the following: annotation of relevant data for extraction, iteration of actions until transitive closure, DOM action simulation, and exposure of presentation style for extraction specification. Automated semantification of text is accomplished by modular integration with QOMPLX's Natural Language Processing (NLP) capability for characterizations such as named entity recognition,

sentiment determination, classification, or even knowledge base construction (often in conjunction with tools like the Q:OS Graphstack knowledge graph service).

At the core of QOMPLX's data extraction system is a very high-throughput web crawling architecture that enables distribution of web crawling load across an arbitrary number of machines for heterogeneous data sources. It is here that crawling intervals and purposes are defined, and diverse post-scraping data cleaning and pre-persistence processing take place before data is registered in Q:OS to track its lineage and utilization across the system and then passed to the rest of the applicable QOMPLX:OS components.

QOMPLX:OS Implementation

Q:Extractor is a specialized microservice that runs on top of Kubernetes as part of Q:OS. It is designed as a multi-tenant capable application that is largely implemented as a Scala-based actor-oriented worker cluster. Extractor recognizes that most web-based collection efforts can be grouped into several related but distinct categories:

- **Simple Crawl**: most standard websites can be viewed and crawled with Scrapy-based "standard" crawling tools since user interactions are relatively limited and surface-level data is the predominant target - this uses an internally developed tool called Crawlee
- **Common Crawl**: Common Crawl is a non-profit organization that exposes data from years of previous crawls to the public and has publicly accessible resultant data stored in major cloud service providers - there is no (or rarely) need to recrawl sites where data is already available in many cases
- **Interactive Crawling**: Rich single page and scripted web applications with extensive server-side interaction can't effectively be scraped by the previously discussed job types. Most commercial tools simply don't attempt to support this kind of capability. QOMPLX's team has world class research experience including inventing OXPath which enables this type of crawling behavior via intuitive but powerful extraction wrappers via XPath extensions. Using a headless browser, we emulate user actions like clicking and form filling, navigation through paginated content, and extraction markers alongside additional useful functions to manipulate the data before it is extracted and passed to Q:OS persistence layers.

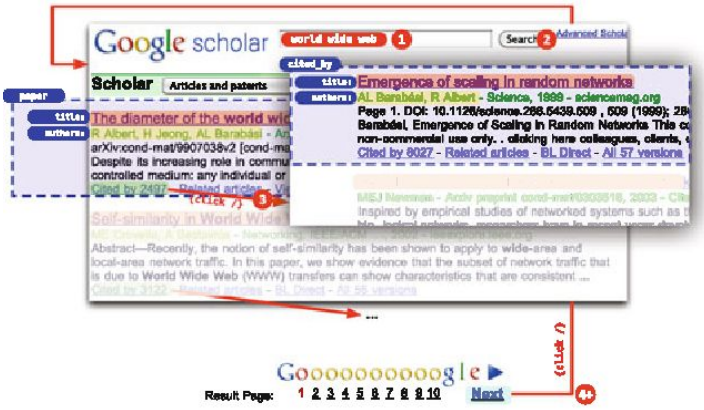
Essential job types:

1. **Simple Crawl:** Simple crawling in Extractor uses Crawlee for scraping requests involving a known data set and high confidence in expected extraction results. It consists of a set of Scrapy-based spiders distributed across a cluster of nodes with real-time queuing controlled by the Redis in-memory data structure store (see Fig. 1). Crawlee works by inputting a set of Uniform Resource Identifiers (URIs), specified by target lists and Regular Expressions (REGEXs) for inclusion and exclusion. Crawlee's DOM Rendering/Interactions module can be used to declaratively specify rich, simulated user interactions, iteration, and targeted extraction during the scraping process. Additionally, the Data Pre/Post Processing Toolkit can be used before or after scraping to refine, filter, and transform extracted data. The Natural Language Processing (NLP) module uses machine learning technology to apply semantification to extracted data for tasks such as sentiment analysis or category classification. Crawlee is launched via a proxy

network that can simulate varied user experiences, such as emulating a user coming from a different country.

Several components within the “standard” distributed scraping framework and common crawl support used by QOMPLX are based on work that was originally funded by the Defense Advanced Research Projects Agency (DARPA) that produced a variety of open-source, distributed web crawling frameworks. These tools extend and simplify development of crawlers leveraging underling tools like Scrapy (with the notable difference that we are not using Kafka and have instead elected to expose a RESTful JSON API and a command line interface (CLI) which leverages the Redis in-memory data structure store in order to coordinate requests among the waiting spider instances).

2. **OXPath:** OXPath allows for multi-way navigation (involving multiple links from the same page) and unbounded navigation sequences (by following links on results pages until there are no longer new links on additional results pages). This capability allows for deeper penetration into any web site to extract all available data. (For example, OXPath can query a restaurant chain’s “Location Finder” utility to collect phone numbers and addresses for every zip code in a single crawl.). It builds a tree of all exposed queries and actions, and then transforms them into a format to be processed by the appropriate components of QOMPLX:OS. OXPath can also be deployed using a proxy network to simulate various user scenarios.




```
doc("scholar.google.com")/
descendant::field()[1]/{"world..."}❶
 /following::field()[1]/{click/}❷
 /(//a[contains(string(.),'Next')]/{click/})*❹
 //div.gs_r:<paper>[.//h3:<title=string(.)>]
   [.//*.gs_a:<authors=substring-before(.,' - ')>]
   [.//a[.~'Cited by']/{click /}❸
 //div.gs_r:<cited_by>[.//h3:<title=.>]
     [.//*.gs_a:<authors=substring-before(.,' - ')>]]
```

3. **Common Crawl:** Common Crawl is a non-profit organization that exposes data from years of previous crawls to the public. A third function of Q:Extractor allows users to define a Q:Extractor job to crawl the data already scraped and available via the Common Crawl repository on Amazon’s S3 storage platform. This provides a more cost-efficient and user-specific solution as an alternative to Crawlee and OXPath.

Scraping processing time is essentially determined by a combination of (a) response time for crawlers to receive a job, (b) the time to execute the actual job, and (c) the degree of effort and time required to perform post-scraping data processing in advance of distribution of job results to a messaging service and/or persistence service.

Ready to learn more about *QOMPLX:OS*? Contact us today.

+1 (703) 995-4199 | info@QOMPLX.com | www.QOMPLX.com

Why QOMPLX®

QOMPLX makes it faster and easier for organizations to integrate all of the disparate data sources across the enterprise into a unified analytics infrastructure to make better decisions. This broader analytics infrastructure is provided through QOMPLX:OS, an enterprise operating system that powers QOMPLX's decision platforms in cybersecurity, insurance underwriting, and quantitative finance. Headquartered in Tysons, VA, QOMPLX, Inc. also has offices in New York and London. More information about QOMPLX can be found at https://www.qomplx.com/.