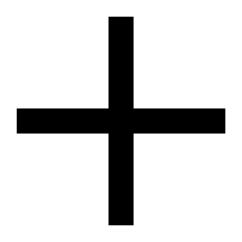
```
▼<api batchcomplete="">
  ▼<query>
    ▼<pages>
      ▼<page _idx="23034" pageid="23034" ns="0" title="Piano">
        ▼<revisions>
          ▶<rev contentformat="text/x-wiki" contentmodel="wikitext"
          </revisions>
        </page>
      </pages>
    </query>
  </api>
```

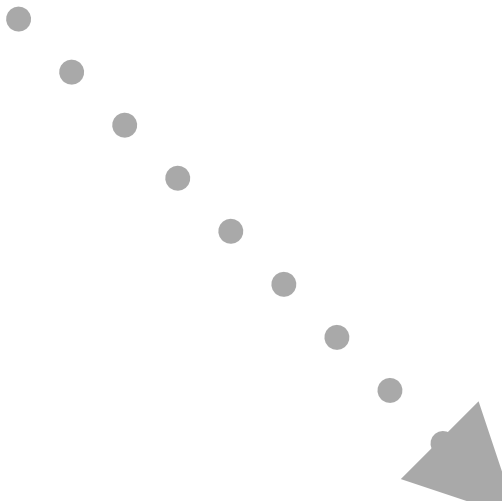

Spark

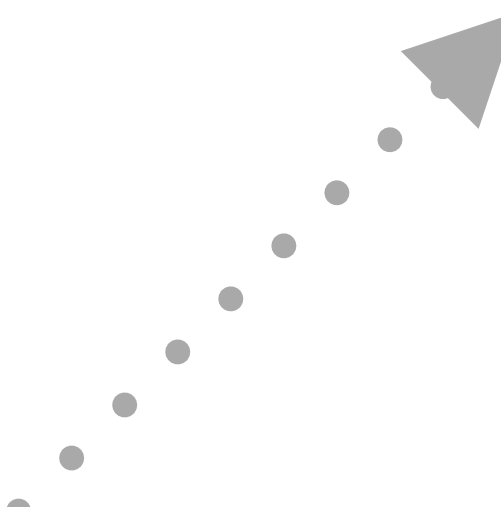

neo4j


elastic

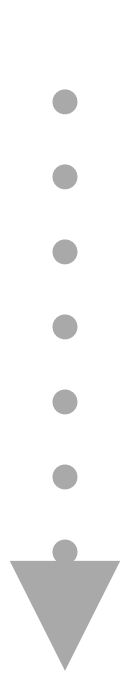
Pageviews

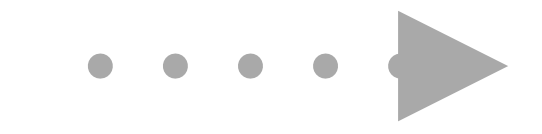







amazon
web services™
S3



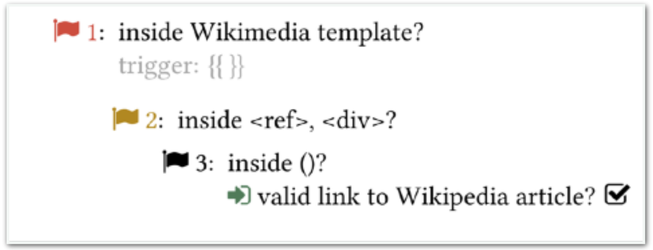


11 million articles


databricks

```
df = sqlContext.read.format('com.databricks.spark.xml')
      .options(rowTag='page').load(full_xml)
```

```
network_df = df.select("title", "revision").map(lambda s:
     (s[0],Article(s[1].text[0]).first_link())).collect()
```




```
class Article:


    def __init__(self, raw_article_text):
        ...

    def check_template(self):
        """check whether inside WikiMedia template: {} """
        ...

    def check_ref_div(self):
        """checks whether inside ref or div"""
        ...

    def check_parenthesis(self):
        ...

    def first_link(self):
        checks...
        if valid_link_to_wiki:
            return first_link
```

11 million articles


```
▼<api batchcomplete="">
  ▼<query>
    ▼<pages>
      ▼<page _idx="23034" pageid="23034" ns="0" title="Piano">
        ▼<revisions>
          ▶<rev contentformat="text/x-wiki" contentmodel="wikitext"
        </revisions>
      </page>
    </pages>
  </query>
</api>
```

+

Pageviews