Automated benchmarks for FIDO, DROID and siegfried
Demo!

• [https://www.itforarchivists.com/siegfried/benchmarks](https://www.itforarchivists.com/siegfried/benchmarks) (Release benchmarks)

• [https://www.itforarchivists.com/siegfried/develop](https://www.itforarchivists.com/siegfried/develop) (Development benchmarks)
iPRES Systems Showcase

A corpus created for the 2014 iPRES conference comprising 2,298 files (656 GB). Represents a range of formats, including AV and some uncommon types. Sourced from http://www.webarchive.org.uk/datasets/ipres.do/

RESULTS

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>sf</td>
<td>Siegfried in its default mode (-multi 1).</td>
<td></td>
</tr>
<tr>
<td>sf mult32</td>
<td>Siegfried with ‘sf mult32 2’ (sf scans multiple files simultaneously).</td>
<td></td>
</tr>
<tr>
<td>sf 64k</td>
<td>Siegfried with ‘sf -multi 32 -sig pronom 64 sig’ (a buffer of 64k).</td>
<td></td>
</tr>
<tr>
<td>sf 128k</td>
<td>Siegfried with ‘sf -multi 32 -sig pronom 128 sig’ (a buffer of 128k).</td>
<td></td>
</tr>
<tr>
<td>droid 64k</td>
<td>DROID in its default mode (a buffer of 64k).</td>
<td></td>
</tr>
<tr>
<td>droid 128k</td>
<td>DROID with FIDO’s default buffer (a buffer of 128k).</td>
<td></td>
</tr>
<tr>
<td>droid nomult</td>
<td>DROID without a buffer limit (a buffer of -1).</td>
<td></td>
</tr>
<tr>
<td>fido 64k</td>
<td>FIDO with DROID’s default buffer (a buffer of 64K).</td>
<td></td>
</tr>
<tr>
<td>fido 128k</td>
<td>FIDO in its default mode (a buffer of 128K).</td>
<td></td>
</tr>
<tr>
<td>fido 1mb</td>
<td>FIDO with Archimedeans’s default buffer (a buffer of 1mb).</td>
<td></td>
</tr>
</tbody>
</table>

The tools differed in output for 370 files in the corpus. 10 of those differences are because of Siegfried’s use of a text identification algorithm and the following chart excludes those files.

<table>
<thead>
<tr>
<th>file</th>
<th>sf</th>
<th>sf 64k</th>
<th>sf 128k</th>
<th>droid 64k</th>
<th>droid nomult</th>
<th>fido 64k</th>
<th>fido 128k</th>
<th>fido 1mb</th>
</tr>
</thead>
<tbody>
<tr>
<td>/root/corpora/ipres-systems-showcase-00000023.bmp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/root/corpora/ipres-systems-showcase-files/04 Cross Diagonal v3d</td>
<td>tmt/443</td>
<td>tmt/443</td>
<td>tmt/443</td>
<td>tmt/443</td>
<td></td>
<td>tmt/443</td>
<td>tmt/443</td>
<td>tmt/443</td>
</tr>
<tr>
<td>/root/corpora/ipres-systems-showcase-files/8800 LBL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/root/corpora/ipres-systems-showcase-files/12.png</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/root/corpora/ipres-systems-showcase-files/3D.jpg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Unzipping

This benchmark checks the `sf-z` command (scans within zip files and other container formats) when run against the iPres corpus.

RESULTS

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>Master branch of github.com/richardlehane/siegfried. Corresponds to latest production release.</td>
<td>2m25.716804378s</td>
</tr>
<tr>
<td>develop</td>
<td>Develop branch of github.com/richardlehane/siegfried. Tip of development and potentially unstable.</td>
<td>2m25.850805317s</td>
</tr>
</tbody>
</table>

The tools differed in output for 0 files in the corpus.

Profile

File: sf-spec
Type: cpu
Time: May 14, 2020 at 2:59am (UTC)
Duration: 30.15s, Total samples = 64.19s (212.88%)
Showing nodes accounting for 61.51s, 95.82% of 64.19s total
Dropped 256 nodes (cum <= 0.32s)

fwac
(*fwac) match
39.48s (61.50%)
of 61.40s (95.65%)

siegreader
(*Reader) ReadByte
18.27s (28.46%)
of 18.37s (28.62%)

...
1. master branch (release benchmarks)
   develop branch (develop benchmarks)

2. Provision server (github.com/richardlehane/provisioner)
   Run jobs (github.com/richardlehane/runner)

3. BACKBLAZE
   Copy corpora
   RCLONE
   packet

4. Read job list (JSON)
   Write job results (JSON)

5. Google Cloud
   Publish on itforarchivists.com
Pros

• Transparent
  • Server specs available on packet.com
  • Code on github inc. server config in provisioner repo
  • Tasks published as JSON on itforarchivists.com
  • Unfiltered results immediately available in raw JSON and views
  • Shows history too

• Automated
  • Not dependent on local setup
  • Doesn’t use local compute
  • Collaborators
  • More that is automated, simpler it is to maintain

• Cheap
• Confidence

Cons

• Fragile
  • Tools can break it
  • Cloud services can break it