This repository | Search   Pull requests   Issues   Gist

google / **battery-historian**
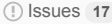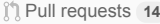
⊙ Watch ▾ 123   ★ Star 1,174   ⑂ Fork 202

<> Code   ⓘ Issues 17   ⎇ Pull requests 14   ⚡ Pulse   ▥ Graphs

Battery Historian is a tool to analyze battery consumers using Android "bugreport" files.

⦿ **18** commits   ⑂ **1** branch   ⬨ **0** releases   **6** contributors

Branch: master ▾   New pull request   New file   Upload files   Find file   HTTPS ▾   https://github.com/goog   ⎘   Download ZIP

kwadkore Fixing some issues in the setup script. …   Latest commit b7db218 2 days ago

📁 activity	Updating Battery Historian.	4 days ago
📁 aggregated	Updating Battery Historian.	4 days ago
📁 analyzer	Updating Battery Historian.	4 days ago
📁 bugreportutils	Updating Battery Historian.	4 days ago
📁 build	External release of Battery Historian 2.0.	10 months ago
📁 checkindelta	Updating Battery Historian.	4 days ago
📁 checkinparse	Updating Battery Historian.	4 days ago
📁 checkinutil	Updating Battery Historian.	4 days ago
📁 cmd	Updating Battery Historian.	4 days ago
📁 csv	Updating Battery Historian.	4 days ago
📁 historianutils	Fixing some issues in the setup script.	2 days ago
📁 js	Fixing some issues in the setup script.	2 days ago
📁 kernel	Updating Battery Historian.	4 days ago
📁 packageutils	Updating Battery Historian.	4 days ago
📁 parseutils	Fixing some issues in the setup script.	2 days ago
📁 pb	Updating Battery Historian.	4 days ago
📁 powermonitor	Updating Battery Historian.	4 days ago
📁 presenter	Updating Battery Historian.	4 days ago
📁 screenshots	Updating README.	4 days ago
📁 scripts	Updating Battery Historian.	4 days ago
📁 sliceparse	External release of Battery Historian 2.0.	10 months ago
📁 static	Updating Battery Historian.	4 days ago
📁 templates	Updating Battery Historian.	4 days ago
📄 LICENSE	Updating Battery Historian.	4 days ago
📄 README.md	Fixing some issues in the setup script.	2 days ago
📄 regen_proto.sh	External release of Battery Historian 2.0.	10 months ago
📄 setup.go	Fixing some issues in the setup script.	2 days ago

▤ **README.md**

Battery Historian

Battery Historian is a tool to inspect battery related information and events on an Android device running Android 5.0 Lollipop (API level 21) and later, while the device was on battery. It allows application developers to visualize system and application level events on a timeline with panning and zooming functionality, easily see various aggregated statistics since the device

was last fully charged, and select an application and inspect the metrics that impact battery specific to the chosen application. It also allows an A/B comparison of two bugreports, highlighting differences in key battery related metrics.

Getting Started

If you are new to the Go programming language:

- Follow the instructions available at http://golang.org/doc/install for downloading and installing the Go compilers, tools, and libraries.
- Create a workspace directory according to the instructions at http://golang.org/doc/code.html#Organization.
- Ensure that `GOPATH` and `GOBIN` environment variables are appropriately set and added to your `$PATH` environment variable. `$GOBIN should be set to $GOPATH/bin`.

  - For Windows, you may set environment variables through the "Environment Variables" button on the "Advanced" tab of the "System" control panel. Some versions of Windows provide this control panel through the "Advanced System Settings" option inside the "System" control panel.

  - For Linux and Mac OS X, you can add the following lines to your ~/.bashrc or ~/.profile files (assuming your workspace is $HOME/work):

    ```
    export GOPATH=$HOME/work
    export GOBIN=$GOPATH/bin
    export PATH=$PATH:$GOBIN
    ```

Next, install Git from https://git-scm.com/downloads if it's not already installed.

Next, make sure Python 2.7 (NOT Python 3!) is installed. See https://python.org/downloads if it isn't, and ensure that python is added to your `$PATH` environment variable.

Next, install Java from http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Next, download the Battery Historian code and its dependencies:

```
$ go get -d -u github.com/google/battery-historian/...
```

Finally, run Battery Historian!

```
$ cd $GOPATH/src/github.com/google/battery-historian

# Compile Javascript files using the Closure compiler
$ go run setup.go

# Run Historian on your machine (make sure $PATH contains $GOBIN)
$ go run cmd/battery-historian/battery-historian.go [--port <default:9999>]
```

Remember, you must always run battery-historian from inside the `$GOPATH/src/github.com/google/battery-historian` directory:

```
cd $GOPATH/src/github.com/google/battery-historian
go run cmd/battery-historian/battery-historian.go [--port <default:9999>]
```

How to take a bug report

To take a bug report from your Android device, you will need to enable USB debugging under `Settings > System > Developer Options`. On Android 4.2 and higher, the Developer options screen is hidden by default. You can enable this by following the instructions here.

To obtain a bug report from your development device:

```
$ adb bugreport > bugreport.txt
```
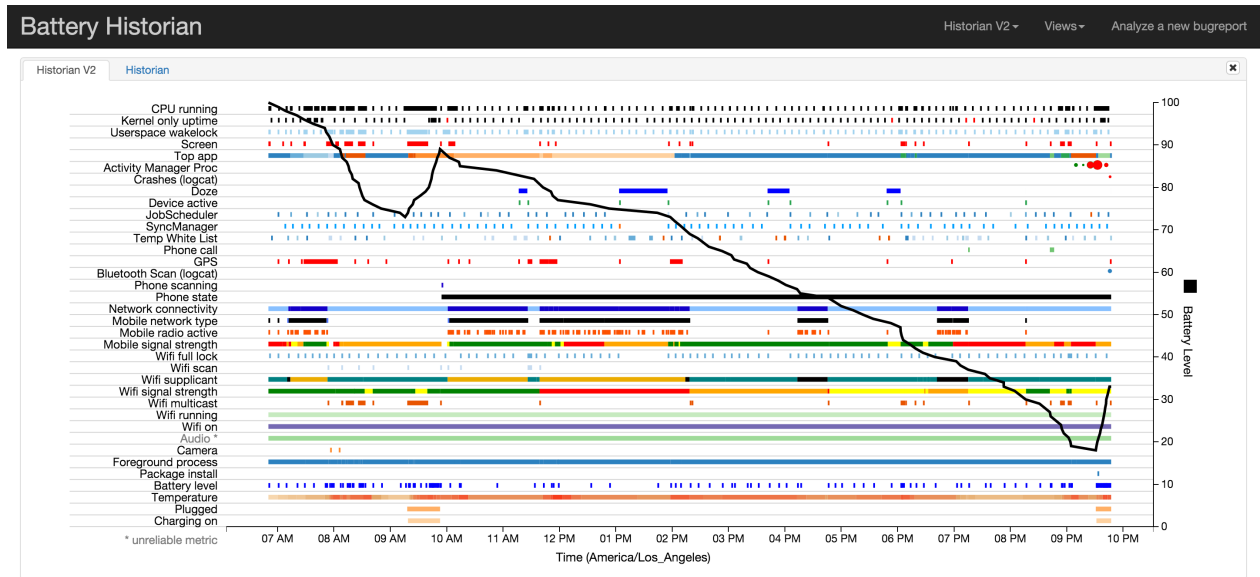
Start analyzing!

You are all set now. Run `historian` and visit http://localhost:9999 and upload the `bugreport.txt` file to start analyzing.

Screenshots

Timeline:

System stats:

App stats:

App Selection

com.google.android.youtube (Uid: 10078)

Tables

- ▸ System Stats
- ▸ History Stats
- ▾ App Stats
  - Misc Summary
  - Network Information
  - Wakelocks
  - Services
  - Process info
  - Sensor Use

System Stats | History Stats | App Stats

Copy

Application	com.google.android.youtube
Version Code	110456640
UID	10078
Device estimated power use	0.27%
Foreground	1 times over 5m 35s 161ms
CPU user time	1m 16s 170ms
CPU system time	48s 45ms
Device estimated power use due to CPU usage	0.03%

— Network Information:

Search: _____ Copy

Mobile data transferred	1.85 MB total (1.77 MB received, 83.16 KB transmitted)
Wifi data transferred	117.61 KB total (102.18 KB received, 15.43 KB transmitted)
Mobile packets transferred	2007 total (1535 received, 472 transmitted)
Wifi packets transferred	243 total (130 received, 113 transmitted)
Mobile active time	1m 47s 289.41ms
Mobile active count	8

+ Wakelocks:

+ Services:

+ Process info:

+ Sensor Use:

Advanced

To reset aggregated battery stats and history:

```
adb shell dumpsys batterystats --reset
```

Wakelock analysis

By default, Android does not record timestamps for application-specific userspace wakelock transitions even though aggregate statistics are maintained on a running basis. If you want Historian to display detailed information about each individual wakelock on the timeline, you should enable full wakelock reporting using the following command before starting your experiment:

```
adb shell dumpsys batterystats --enable full-wake-history
```

Note that by enabling full wakelock reporting the battery history log overflows in a few hours. Use this option for short test runs (3-4 hrs).

Kernel trace analysis

To generate a trace file which logs kernel wakeup source and kernel wakelock activities:

First, connect the device to the desktop/laptop and enable kernel trace logging:

```
$ adb root
$ adb shell

# Set the events to trace.
$ echo "power:wakeup_source_activate" >> /d/tracing/set_event
$ echo "power:wakeup_source_deactivate" >> /d/tracing/set_event

# The default trace size for most devices is 1MB, which is relatively low and might cause the logs to overf
# 8MB to 10MB should be a decent size for 5-6 hours of logging.

$ echo 8192 > /d/tracing/buffer_size_kb

$ echo 1 > /d/tracing/tracing_on
```

Then, use the device for intended test case.

Finally, extract the logs:

```
$ echo 0 > /d/tracing/tracing_on
$ adb pull /d/tracing/trace <some path>

# Take a bug report at this time.
$ adb bugreport > bugreport.txt
```

Note:

Historian plots and relates events in real time (PST or UTC), whereas kernel trace files logs events in jiffies (seconds since boot time). In order to relate these events there is a script which approximates the jiffies to utc time. The script reads the UTC times logged in the dmesg when the system suspends and resumes. The scope of the script is limited to the amount of time stamps present in the dmesg. Since the script uses the dmesg log when the system suspends, there are different scripts for each of the device, with only difference being the device specific dmesg log it tries to find. These scripts have been integrated into the Battery Historian tool itself.

Powermonitor analysis

Powermonitor files should have the following format per line:

```
<timestamp in epoch seconds> <amps>
```

Entries from the powermonitor file will be overlaid on top of the timeline plot.

To ensure the powermonitor and bug report timelines are somewhat aligned, please reset the batterystats before running any powermonitor logging:

```
adb shell dumpsys batterystats --reset
```

And take a bug report soon after stopping powermonitor logging.

If using a Monsoon:

Download the AOSP Monsoon Python script from
https://android.googlesource.com/platform/cts/+/master/tools/utils/monsoon.py

```
# Run the script.
$ monsoon.py --serialno 2294 --hz 1 --samples 100000 -timestamp | tee monsoon.out

# ...let device run a while...

$ stop monsoon.py
```

Modifying the proto files

If you want to modify the proto files (pb/*/*.proto), first download the additional tools necessary:

Install the standard C++ implementation of protocol buffers from
https://github.com/google/protobuf/blob/master/src/README.md

Download the Go proto compiler:

```
$ go get -u github.com/golang/protobuf/protoc-gen-go
```

The compiler plugin, protoc-gen-go, will be installed in $GOBIN, which must be in your $PATH for the protocol compiler, protoc, to find it.

Make your changes to the proto files.

Finally, regenerate the compiled Go proto output files using `regen_proto.sh` .

Other command line tools

```
# System stats
$ go run cmd/checkin-parse/local_checkin_parse.go --input=bugreport.txt
```

```
# Timeline analysis
$ go run cmd/history-parse/local_history_parse.go --summary=totalTime --input=bugreport.txt

# Diff two bug reports
$ go run cmd/checkin-delta/local_checkin_delta.go --input=bugreport_1.txt,bugreport_2.txt
```

Support

- G+ Community (Discussion Thread: Battery Historian):
  https://plus.google.com/b/108967384991768947849/communities/114791428968349268860

If you've found an error in this sample, please file an issue: https://github.com/google/battery-historian/issues

License

Copyright 2016 Google, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.