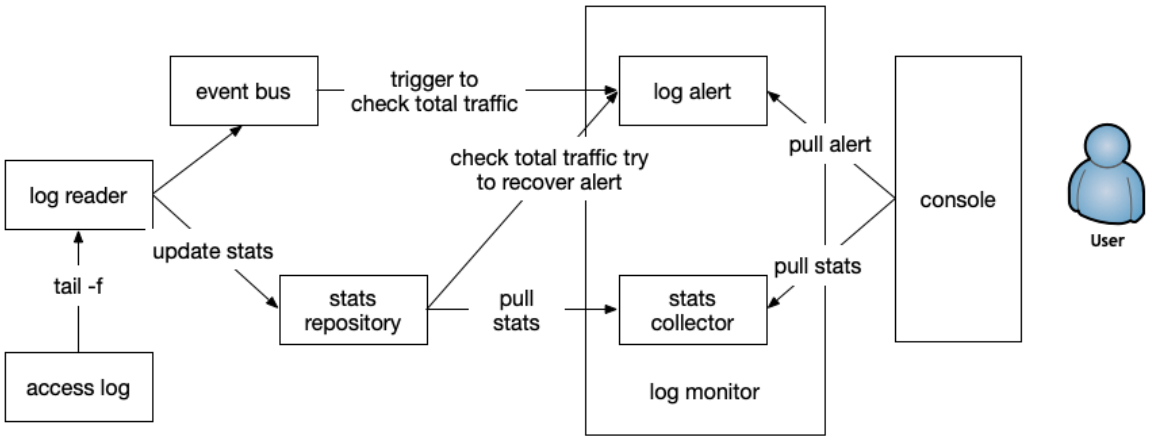
DatadogTask

This Java project is a take home-task for Datadog interview. The application continuously monitors a CLF HTTP access log and provides two features.

1. It displays stats every n second about the traffic in the last n seconds. The stats include: total requests, QPS(request per second), top k visited sections, top k client IPs, top k users.
2. Whenever the number of the total request in the last n seconds exceeds a threshold, the application displays an alert. Whenever the number of the total request drops below the threshold, it displays a recovered message.



High level view of components

Package and run

Requires Java 8+, Maven 3+

1. Package

```
mvn clean -Dmaven.test.skip=true package
```

2. Run

```
java -jar ./target/DataDogTask-1.0-SNAPSHOT-jar-with-dependencies.jar [/path/config.properties]
```

Note: It is not necessary to have a log file in place before running the application. The application will pick up the file automatically once it is created.

3. Access log generator

There is also a simple access log generator in the project. In case you want to generate the log, you can run it.

```
java -cp ./target/DataDogTask-1.0-SNAPSHOT-jar-with-dependencies.jar com.datadog.task.LogGenerator [/tmp/access.log] [10]
```

4. Logging setup

The jar is bundled with Logback, you may specify the location of the default configuration file with a system property named `logback.configurationFile`.

Supported properties

properties

key	defaut value	description
logfile.path	/tmp/access.log	The access log full path.
alert.threshold.qps	10	The QPS threshold to trigger hight traffic alert.
time.window.sec	120	The length of time window in second. The application holds data for this period of time.
statistics.interval.sec	10	The time interval for reporting stats.
topk	5	The top k frequently visited items.

Datastructure explained

Internally, the stats repository uses an array with a length of time.window.sec as a ring buffer, the ith slot stores the stats for the n second where i = n % array.length. For checking the total requests, the complexity is constant regardless of how many requests are received so far.

Things can be improved

1. The log reader utilizes the Tailer from apache.common.io, it basically tries to read the new content of the log at a fixed interval. The solution is simple but with some drawbacks. If there is no new content of the log, the reader thread will wake anyway. If the interval is short, it consumes too much CPU, if the interval is long, there will be a lag between the log file and the reader.

   How to improved:

   Linux provides inotify which can be used to monitor the file change, but it also has some drawbacks. In production, people usually use loop polling combined with inotify to build a log agent.

2. The application uses an hashmap to stores the frequencies for each item, such as section, client IP, et.. This hashmap is unbouned, so if the log contains logs of unique sections or client IPs, the size of the hashmap may be a problem.

   How to improved:
   Limited the size of the hashmap to control the memory usage.

3. If the application crash and restart, it will read the log from start again. It is not an issue for this task because the application only stores the log which generated in last n seconds. But in real life, log is usually shipped by the agent to some place else. Reading from the start may cause lots of duplicated log resent.

   How to improved:
   Instead of storing the read position in memory as the Apache Tailer does, a production-ready log agent usually persists the read position into a local file. Whenever the log agent restarts, it will start shipping the log from the last read position.

4. The console is based on Lanterna, it is not easy to test a GUI application and the layout is not very flexible. For example, if the topk is very large, the console may not display correctly.

Limitation

According to Common Log Format sepcifcation, the month is the log is recorded as abbreviated month name, e.g. Oct(locale dependent). However, this project only supports **Locacle.US**.

Attributes

- A Beginner's Guide to Logstash Grok
- Apache file tailer
- Text-Based GUI with Lanterna in Java