

Preferred Networks インターン選考 2018 コーディング課題バックエンド分野

変更履歴

- 2018-5-1: 初版

回答にあたっての注意

- (特に注釈がない限りは) 各言語の標準ライブラリの関数のみを使用してください。
- ソースコードには以下のいずれかの言語を使用してください。
 - C, C++, Python, Go, Scala, Rust, Java, C#, Ruby
 - 複数の言語を使用しても構いません。
- 課題には自分だけで取り組んでください。この課題を他の応募者を含めた他人と共有・相談することを禁止します。漏洩の証拠が見つかった場合、その応募者は失格となります。ある応募者が別の応募者に回答をコピーさせた場合、双方の応募者が失格となります。
- 想定所要時間は2日です。全課題が解けていなくても提出は可能ですので、学業に支障の無い範囲で取り組んで下さい。

提出物

- 課題のプログラムのソースコード
- 実行方法を示したレポート
 - プログラム開発に使用した OS・プログラミング言語・バージョンを記述してください
 - プログラムのビルドや、実行に必要な Makefile 等のコマンドを添付・記述してください

評価基準

以下のような要素を評価します。

- 出力された結果が正しいこと。
- 実行速度が速いこと。
- 使用メモリ量が少ないこと。
- ソースコードが他人が読んで理解しやすいものになっていること。
- ソースコードが正しいものであることを検証するために、単体テストや検証のためのコードが書かれていること。

- 提出物を見て追試がしやすい形になっていること。
- レポートが要点についてわかりやすくまとまっていること。

提出方法

上記の提出物を単一のパスワード無し zip ファイルにまとめ、[こちら](#)の専用フォームより応募してください。締切は日本時間 2018 年 5 月 14 日（月）23:59 です。

問い合わせ

問題文に関して質問がある場合は intern2018@preferred.jp までご連絡ください。問題文に訂正が行われた場合は応募者全員にアナウンスいたします。なお、アプローチや解法に関する問い合わせにはお答えできません。

問題文

ログの分析ツールを作成してください。分析処理は静的なログファイルを対象とし、オンライン処理やストリーム処理を前提とする必要はありません。各問題とも以下の 2 つのデータを対象とします。

- [log_s.zip](#) (MD5: 7976df39e96fe03a1a351ee95193ffd8)
- [log_l.zip](#) (MD5: 68545fd4ce6a3823dbe56318c6a895df)

ダウンロード後、適当な名前に変更してください。

```
$ wget https://preferredjp.box.com/shared/static/mc01xtkcn0qmdgbb1r53uzljeva6e9tq.zip -O log_s.zip
$ wget https://preferredjp.box.com/shared/static/gpmix7flrrl4badutdqwo4hr9q2xnasp.zip -O log_l.zip
```

データは複数のディレクトリで構成され、以下のように、それぞれのディレクトリにサーバー群のログデータが含まれます。

```
log_l
|- server100
    |- app.log.0 <- oldest log file in "server100" directory
    |- app.log.1
    |- ...
    |- app.log <- newest log
```

ログには以下の項目が含まれています

- ログ種別: DEBUG / INFO / WARNING / ERROR
- 時刻: 日付+時刻+タイムゾーン
- サーバー名 (1つのログに複数のサーバーが含まれている場合があります)
- ユーザー名
- ログ本文 (以下のいずれかが含まれます)
 - CPU 使用率 (0.1=10%)
 - エラー内容およびスタックトレース
 - その他

問題 1

CPU 使用率が 95%を超えている時刻およびそのサーバー名を出力してください。95%という閾値は起動引数などでプログラムに外部から与えられるようにしてください。

問題 2

全サーバーの CPU 使用率の 5 分間中央値を 1 分毎に、表示してください。表示形式は自由ですが、ヒートマップや wireframe surface 等を使用し、1 つにまとまっているとよいです。可視化処理および画像出力処理はライブラリを使用しても構いません。

例

- サーバー 1 2018-04-01 10:00:00~10:05:00 の中央値 50%
- サーバー 1 2018-04-01 10:01:00~10:06:00 の中央値 55%
- サーバー 1 2018-04-01 10:02:00~10:07:00 の中央値 40%
- ...

この場合、サーバー 1 の軸に {2018-04-01 10:05:00,50%}, {2018-04-01 10:06:00,55%}, {2018-04-01 10:07:00,40%} をプロットします。

問題 3

問題 2 で出力した CPU 使用率の 5 分間中央値が 90%を超えている場合の、その 5 分間で出力された WARNING ログおよび ERROR ログの本文を全て出力してください。表示形式は自由です。

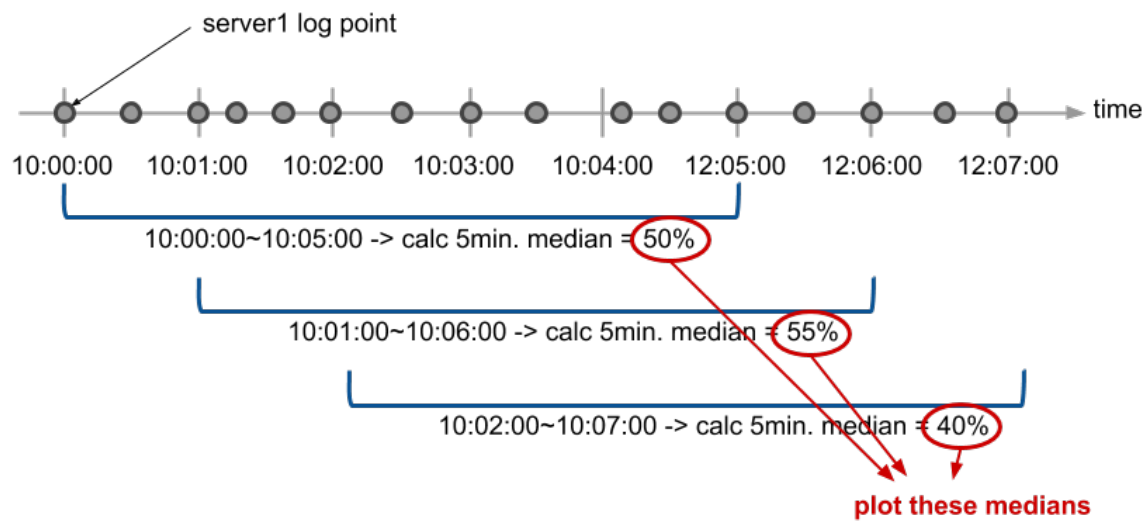


Figure 1: how_to_calc_median

例

- サーバー 2 2018-04-01 10:00:00~10:05:00 の中央値 33%
- サーバー 2 2018-04-01 10:01:00~10:06:00 の中央値 91%
- サーバー 2 2018-04-01 10:02:00~10:07:00 の中央値 95%
- サーバー 2 2018-04-01 10:03:00~10:08:00 の中央値 88%
- ...

この場合、[10:01:00~10:06:00] および [10:02:00~10:07:00] に出力された WARNING ログ および ERROR ログの本文を出力します。[10:02:00~10:06:00] は重なっていますが、同じログは複数回出力されないことが望ましいです。

問題 4 (自由回答)

※学業に支障の出ない範囲で取り組んでください。

課題 1~3 の結果を HTTP で登録する HTTP クライアントおよび、その結果に対して、サーバー名・時間帯などを指定して取得する HTTP サーバーを実装してください。ライブラリは自由に使用して構いません。API の形式は REST / GraphQL 等、自由とします。

例

例えば下記に例示されているエンドポイントが実装されたサーバーを作成し、起動します。

```
$ ./my_server start -p 8080
```

各課題で出力したファイルを登録する、クライアントを作成します。DB に直接登録するのではなく、HTTP 経由で登録してください。

```
$ ./my_client register task1_result.txt
$ ./my_client register task2_result.txt
$ ./my_client register task3_result.txt
```

“server101”、[2018-04-01 00:00:00 ~ 00:10:00] に CPU 使用率が 95 % 超えた時刻の出力例。

```
$ curl localhost:8080/api/cpu_util/over=95&server=101&from=1522508400&to=1522509000
{
  server: "server101"
  timestamps: [2018-04-01T00:XX:XX.XXX, 2018-04-01T00:YY:YY.YYY, ...]
}
```

“server101”、[2018-04-01 00:00:00 ~ 00:10:00] における CPU 使用率 5 分間中央値のリストの出力例。

```
$ curl localhost:8080/api/cpu_util/medians_5min/server=101&from=1522508400&to=1522509000
{
  server: "server101"
  medians: [0.1, 0.11, ...]
}
```

“server101”、[2018-04-01 00:00:00 ~ 00:10:00] における CPU 使用率 5 分間中央値が 90 %を超えた場合の、エラーログの出力例。

```
$ curl localhost:8080/api/logs/level=error&median_over=90&server=101&from=1522508400&to=1522509000
{
  server: "server101"
  logs: [
    {
      time: "2018-04-01T00:XX:XX.XXX",
      level: "ERROR",
      message: "error message"
    }
  ]
}
```

これらエンドポイント例および出力例に従う必要はありません