### **Projektdokumentation Log-Datei-Visualisierung**

Update: Nach der Addition des aktuellen Features kann das

Das Log-Datei-Visualisierungsprogramm (im Folgenden „Plotter“ genannt) dient der Visualisierung des zeitlichen Verlaufs des Verbrauchs eines oder mehrerer Jobs auf dem Lichtenberg-Hochleistungsrechner. Es generiert einen Graphen, auf dem die akkumulierten Kernstunden sowie eine monatsweise Kategorisierung abzulesen sind. Im unteren Graphen wird für jeden einzelnen Job ein Punkt generiert, der die Effizienz des jeweiligen Jobs repräsentiert.

**Inhalte:**

* Quota\_Visualisierer.py
* Time\_functions.py
* Parsing.py
* Drawing.py
* Pillow.py
* Nutzer\_Dokumentation\_Visualisierung.docx
* Projektdokumentation\_Quotavisualisierung.docx

**Anforderungen:**

Das Projekt nutzt Python (Version 3.6) *Numpy*, *Pyplot und* *Matplotlib,* das Modul *Pillow* wird verwendet, befindet sich aber auch im Repository und wird mit gepullt.

Eine weitere Installation ist nicht notwendig.

**Anwendung**

*Schritt 1*: Geben Sie in ein Terminal Ihrer Wahl folgenden Befehl ein:

|  |
| --- |
| * python Quota\_Visualisierer.py Source\_Datei Ziel\_Datei q=Jahresquota  –s=Startpunkt –p=Projektname |

Hierbei steht „Source\_Datei“ für eine CSV-Datei, die das Zeichen „*|*“zur Trennung verwendet, sie entspricht einer SLURM-Log-Datei.  
*Ziel\_Datei* ist der Name unter dem der Graph gespeichert werden soll.  
*Jahresquota* bezeichnet eine Menge an Kernstunden, die als Richtwert für ein Jahr verwendet wird  
Der Begriff Startpunkt wird mit einem Datum oder einem Datum inklusive sekundengenaue Uhrzeit in folgendem Format angegeben: „*YYYY-mm-dd-HH-MM-SS*“.  
Falls nur ein Datum angegeben wird (*YYYY-mm-dd*), verwendet der Plotter die erste Sekunde des jeweiligen Tags als den genauen Zeitpunkt.  
Der Projektname ist eine Filterfunktion, wird ein Name gesetzt, so werden nur Jobs berücksichtigt, bei denen der Name Teil des Projektnamens ist.   
Als Zieldatei können alle üblichen Bildformate und auch Dokumentenformate wie „.pdf“ verwendet werden.

|  |
| --- |
| python C:\Folder\Quota\_Visualisierer.py C:\Folder\Mueller.log-example  -o=Abbildung3.png –q=10000.-s=2017-11-07 -p=005 |

Die Parameter Jahresquota (*-q=)*, Startpunkt *(-s=)* sowie Projektname *(-p=)* sind optional.  
Das Script generiert unter dem von Ihnen angegebenen Namen, eine Grafik, die den Rechenzeitverbrauch der Projekte visualisiert und das Verhalten in Relation zur Jahresquota farblich markiert (hellgrüne Bereiche, in denen der Verbrauch unter 70% des Vergleichswertes liegt, dunkelgrüne, in denen er sich zwischen 70% und 110% befindet, sowie den gelben Bereichen, welche einen Verbrauch zwischen 110% und 150% signalisieren. Darüber hinaus existieren rote Markierungen, welche einer Corerechenzeit von mehr als 150% des Monatsquotas entspricht.).  
Jeder Abschnitt repräsentiert einen Kalendermonat, mit dem 1. des Monats beginnend.  
Ein Funktionsaufruf könnte ein Output wie die Folgende Grafik erzeugen:

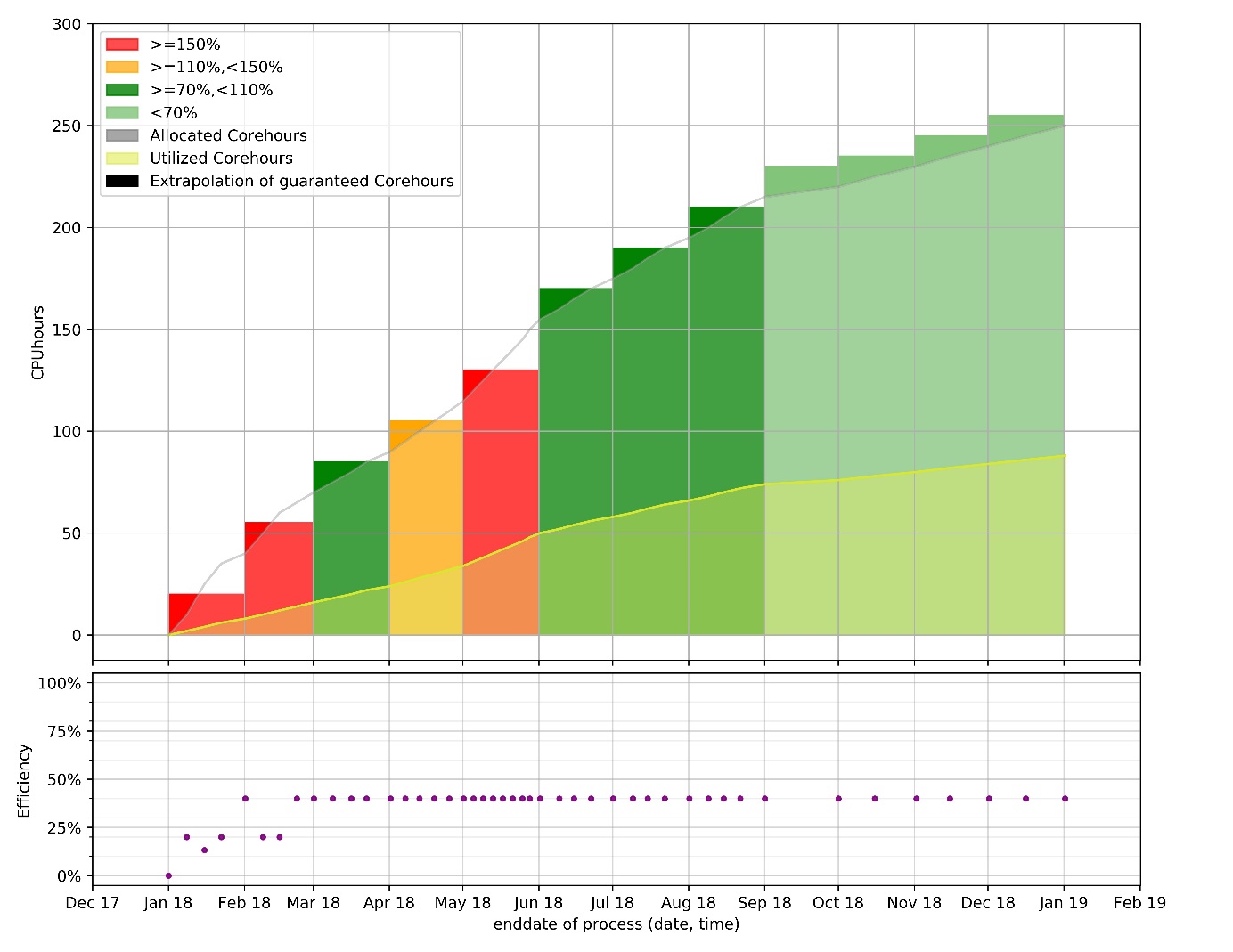


Abbildung : Eine Visualisierung einer Beispiel-Log-Datei.

Der Plotter erstellt einen Graphen ähnlich dem in Abbildung 1. Hier ist in der oberen Linie die verwendete Rechenzeit in Corestunden gegenüber dem Endzeitpunkt des Prozesses dargestellt. Somit lässt sich der zeitliche Verlauf des Rechenleistungsverbrauchs interpretieren, hier durch ein farbiges Highlight der möglichen Quotas.  
Als Monatsquota gilt hier ein Zwölftel des als Parameter eingegebenen Jahresquotas.

**Erklärungsbedürftige** **Funktionen**:

**translate\_date\_to\_sec**: Diese Funktion dient dem umwandeln von *datetime*-Formaten in *timestamps*, die pure Integer sind. Darüber hinaus dient die Funktion als Filter für nicht auswertbare Daten, die als Text angegeben sind. So wird zum Beispiel der Endzeitpunkt von Jobs, die noch ausgeführt werden als *Unknown* angezeigt.

**Globale** **Variablen**:

**thresholds:**

Ein Array von Float-Werten, die das Verhältnis der jeweiligen Farbbereiche beschreiben

**colors:**

Ein Array von Strings mit Farbcodes, welche in den Bereichen verwendet werden, die in thresholds beschrieben werden.

**maximum:**

Ein Farbcode, der verwendet wird, falls der höchste threshold überschritten wird. (in dem Beispiel aus Abb.1 war dies die Farbe Rot)