Skalierungs-Visualisierungsprogramm

Anforderungen:  
Das Skalierungs-Visualisierungsprogramm benötigt einen Rechner mit installierter Python Version, inklusive numpy und matplotlib.

Einsatz:  
Das Skalierungs-Visualisierungsprogramm dient der Veranschaulichung der Skalierbarkeit eines Programms über einen vom Nutzer bestimmten Bereich von Rechenkernen.

Funktionen:

Das Skalierungs-Visualisierungsprogramm (im Folgenden „Reader“ genannt,) liest eine Datei und zeichnet den Graphen einer Modellfunktion, welche darin beschrieben ist in eine Zieldatei, Die Funktion wird in der Quelldatei als „model: “ bzw. „Model: “ beschrieben.

Das Modell besteht aus der Summe einer beliebigen Menge aus Termen in folgender Form:

* **c1\*(p^i)\*log2^j(p)**

Hierbei sind (p^i) und log2^j(p) nicht zwingend erforderlich.

Der Reader bildet die vorgegebene Funktion auf die variable p in einem Bereich ab, der vom User über Parameter vorgegeben wird, und dann auf Zweierpotenzen abgerundet bzw. für die obere Grenze aufgerundet wird.

D.h. wenn der User als Start- und Endpunkt 3 und 555 angibt, verwendet der Reader die Grenzen 2 und 1024.

Falls der Startpunkt kleiner als 1 ist, wird 1 als Startpunkt verwendet.

Im Falle, dass der Endpunkt nicht mindestens eine Zweierpotenz ist, die größer als der Startpunkt ist, wird die dem Startpunkt nächste (höhere) Zweierpotenz verwendet.

Optional kann der Nutzer über einen Fokuspunkt einen Bereich angeben, hierfür werden bis zu 10% des Graphen (10% der x-Achse), zentriert um den gegebenen Punkt farblich hervorgehoben.

In der gelesenen Datei stehen Messwerte, diese Messwerte werden (falls möglich) inklusive der Standardabweichung zu dem Graphen hinzugezeichnet.

Aus den Funktionswerten innerhalb der gegebenen Grenzen generiert das Programm einen Graphen, welcher als Bilddatei unter dem vom User angegebenen Pfad („-o=“) gespeichert wird.

Leistungen:  
Der Reader ist dazu bestimmt, Graphen über 0 für Kernzahlen >= 1 auszugeben. Ergebnisse außerhalb dieser Grenzen sind undokumentiertes Verhalten.

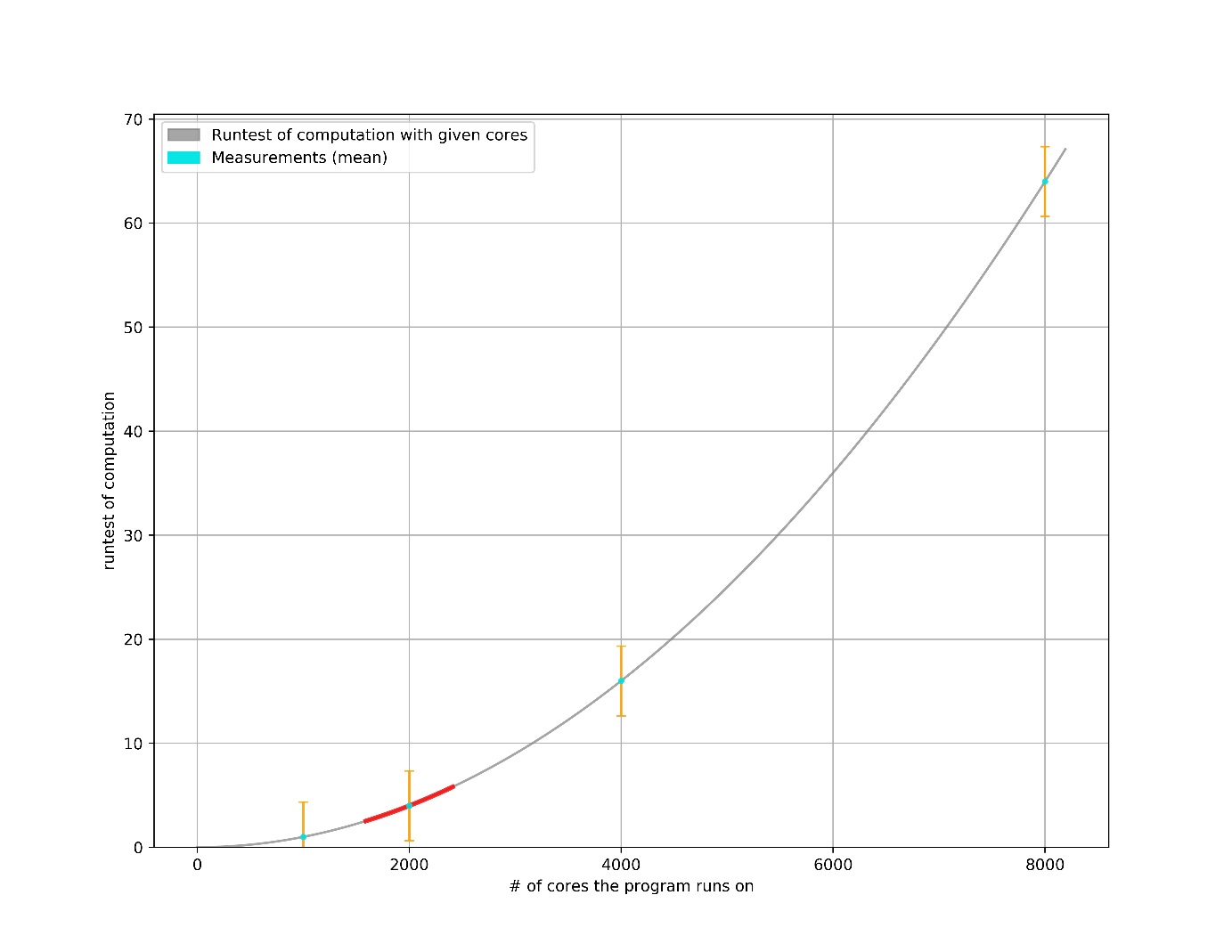
Beispiel:x

Abbildung 1 Der hier gezeigte Graph wurde mit der Funktion f= 0+9.9998\*10^-7\*p^2 generiert, der Bereich rund um 2000 Kerne wurde rot markiert. Zusätzlich sind in Türkis Durchschnitts-Messpunkte eingezeichnet, zum Beispiel am Punkt (16000|256), um diesen Punkt ist die Standardabweichung visualisiert. Die Achsenbeschriftung wird teilweise durch due eingelesene Datei generiert. In diesem Beispiel stehen die Messwerte nicht für Zeit oder Effizienz, sondern für „test“.

Tests:

Innerhalb des Repositories befindet sich eine Datei, die, falls in den selben Ordner wie „reader.py“ gelegt, verschiedene Daten in einen Unterordner generiert, dazu zählen Input und Output Daten.