Erklärung der Vorgehensweise zur Optimierung der Performance

Wann wurde wie entschieden wie viele Threads gestartet werden?

Die Anzahl der gestarteten Threads wurde aufgrund der vorhandenen Hardware entschieden. Dabei wurde eine Methode aufgerufen, die die maximal gleichzeitig unterstützten Threads des Systems ausgibt. Diese wurde am Beginn aufgerufen und als Maximum gesetzt. Dazu wurde folgende Methode verwendet:

```
std::thread::hardware_concurrency()
```

Warum wurde was wie gemacht?

Erster Ansatz

Im ersten Ansatz sollten zuerst alle Pfade zu den Dateien gesammelt werden. Danach sollten diese Pfade auf die Anzahl der Threads aufgeteilt werden und somit eine möglichst schnelle parallele Abarbeitung erfolgen. Dieser Ansatz wurde allerding verworfen, mit dem Hintergrund, dass der parallele Zugriff auf Daten auf der Festplatte langsamer ist, als der Zugriff eines Threads auf die Dateien nacheinander.

Zweiter Ansatz

Im Zweiten Ansatz soll der Main Thread die Dateien aufrufen. Danach werden die Dateien an einen Thread Pool übergebenen. Der Thread Pool startet die vorher festgelegte Anzahl an Threads und zählt die Anzahl der Einser und Nullen aus den bereits geladenen Dateien. Dadurch Erfolgt der Zugriff auf die Festplatte aus nur einem Thread Und dieser öffnet eine Datei Nach der anderen. Da der Flaschenhals der Applikation die Festplatte ist, ist diese voll ausgelastet und die Berechnung erfolgt in eigenen Threads.

Aufrufen/Öffnen der Dateien

Das Öffnen der Dateien wurde mithilfe der Boost Library bewerkstelligt. Dabei wurde das File Mapping verwendet, um möglichst schnellen Zugriff auf Dateien zu erhalten.