

In den folgenden Aufgaben werden Sie das in den Materialien bereitgestellte Programm analysieren und optimieren. Modifizieren Sie das Programm zuerst so, dass nach jedem `write`-Aufruf eine Synchronisation mittels `fsync` ausgeführt wird. Analysieren Sie nun das Programm hinsichtlich seiner Leistung unter Zuhilfenahme der folgenden Werkzeuge.

## 1 iostat (30 Punkte)

Lassen Sie sich während des Programmlaufs die Statistiken mittels `iostat` ausgeben. Beschreiben Sie ausführlich, was sie in der Ausgabe sehen können. Lassen Sie sich anzeigen, wie viele MB pro Sekunde geschrieben werden und vergleichen Sie diesen Wert mit Ihrer Ausgabe. Protokollieren Sie die Ergebnisse und erklären Sie eventuelle Abweichungen.

## 2 VampirTrace-Analyse und Optimierung (150 Punkte)

In dieser Aufgabe soll das auf dem Cluster verfügbare Leistungsanalysewerkzeug Vampir<sup>1</sup> verwendet werden, um das Laufzeitverhalten des Programms auszuwerten. Gehen Sie dazu wie folgt vor:

1. Kompilieren Sie das Programm mit dem VampirTrace-Compiler-Wrapper `vtcc`.
2. Führen Sie das Programm aus. Hierbei entstehen so genannte Spurdaten (Traces) für die spätere Visualisierung; um zusätzlich die E/A-Operationen aufzuzeichnen, exportieren Sie die Umgebungsvariable `VT_IOTRACE_EXTENDED=yes`.
3. Visualisieren Sie die Spurdaten in Vampir (`vampir checkpoint.otf`). (Hinweis: Hierfür können Sie das auf dem Cluster installierte X2Go verwenden, das eine deutlich höhere Leistung als das übliche X-Forwarding bietet.)

Beschreiben Sie nun die wesentlichen Merkmale des Programms und legen Sie Ihren Beschreibungen aussagekräftige Screenshots bei. Gehen Sie dabei besonders detailliert auf das E/A-Verhalten ein. Bewerten Sie dieses und begründen Sie Ihre Meinung.

Überlegen Sie sich nun, wie Sie das Programm hinsichtlich der E/A-Leistung verbessern können. Implementieren und erläutern Sie ihre Optimierungen (achten Sie besonders auf gute Dokumentation).

## 3 Leistungsvergleich (60 Punkte)

Werten Sie nun das Programm mittels aussagekräftigen Messungen aus. Nehmen Sie dabei Messungen für die folgenden Varianten vor:

1. Das Originalprogramm,
2. das Originalprogramm mit Synchronisation

---

<sup>1</sup><http://www.vampir.eu/>

3. und das von Ihnen optimierte Programm.

Verwenden Sie dazu 1–12 Threads und wählen Sie eine geeignete Iterationszahl aus, wobei die Mindestlaufzeit jeweils 120 Sekunden betragen soll. (Hinweis: Für die unterschiedlichen Messungen können unterschiedliche Iterationszahlen gewählt werden.) Visualisieren Sie Ihre Ergebnisse in geeigneten Diagrammen und schreiben Sie jeweils Ihre Erkenntnisse auf.

## **Abgabe**

Erstellen Sie in dem Verzeichnis mit ihren C-Programmen eine Datei `antworten.pdf`. Packen Sie ein komprimiertes Archiv (`.tar.gz`) aus dem sauberen Verzeichnis (ohne Binärdateien).

Senden Sie das Archiv per E-Mail an `hea-abgabe@wr.informatik.uni-hamburg.de`.