Praktikum 5: Testing and Performance Analyse

M.Thaler, 2/2016, ZHAW

1 Einführung

Testing und Performance Analyse sind wichtige Schritte bei der Implementation paralleler Anwendungen und wie in der Vorlesung gezeigt nicht ganz einfach.

In der Testing Phase wird verifiziert, ob das Programmverhalten unter möglichst vielen verschiedenen Randbedingung korrekt arbeit. Die Performance Analyse gibt darüber Aufschluss, wie gut eine Applikation parallelisiert ist, wo sich noch Verbesserungspotential findet und wie sich verschiedene verschiedene Algorithmen verhalten.

In diesem Praktikum werden Sie stellvertretend für andere Werkzeuge das Verfikationstool **valgrind** und den OpenMP Profiler **ompP** kennenlernen. Als Anwendungsbeispiel verwenden wir die Brute Force Methode zum Finden von Primzahlen.

2 Testing

Im Verzeichnis `testperf/a0` haben wir das OpenMP-Programm für die Berechnung der Primzahlen bereitgestellt. Primzahlen im Bereich ≥ 3 lassen sich in zwei Klassen einteilen: Primzahlen der Form $4n + 1$ und der Form $4n + 3$. Das Programm zählt im Bereich `3...MAX_NUM` die gesamte Anzahl Primzahlen `count`, sowie die Anzahl in den beiden Klassen `n41count` und `n43count`.

Verifizieren Sie das Programm und korrigieren Sie ev. Fehler. Korrekte Resultate erhalten Sie wenn Sie die OpenMP Direktiven auskommentieren. Bei der Fehleranalyse kann das Tool `valgrind` wichtige Hinweise liefern (Variante 2 unterdrückt Meldunen bzgl. `libgomp`):

```
valgrind --tool=helgrind --log-file=log.txt ./main.e
valgrind --tool=helgrind --suppressions=./do.supp --log-file=log.txt ./main.e
```

3 Performance Analyse mit ompP

3.1 Installation von ompP

Mit den Praktikumsunterlagen erhalten Sie das File `ompp-0.7.1.tgz`. Entpacken Sie das File und wechseln Sie ins Verzeichnis `ompp-0.7.1`. Editieren Sie `Makefile.defs` und setzen Sie:

```
INSTDIR = $(HOME)/ompp
OMPCC   = gcc
OMPFLAG = -fopenmp
```

Geben Sie nun `make` und anschliessend `make install` ein.

Damit unsere `makefiles` korrekt arbeiten, müssen Sie Ihren Pfad entweder in `.bash_profile` oder in `.bashrc` mit dem von Ihnen gewählten Verzeichnis ergänzen, zum Beispiel:

```
PATH="$PATH:${HOME}/ompp/bin"
export PATH
```

Damit der neue Pfad aktiviert wird, loggen Sie sich am besten aus und wieder ein.

Hinweis:
Wenn Sie auf die Installation von ompP verzichten möchten, können Sie auch auf *Dove* arbeiten, ompP ist dort schon installiert.

3.2 Aufgaben

Ausführliche Informationen zu ompP finden Sie im beigefügten User Guide and Manual. Unten stehend eine Zusammenfassung der wichtigsten Schritte.

3.2.1 Programme mit dem Profiler übersetzen

Im Verzeichniss `testperf/a1` haben wir ein entsprechendes `makefile` vorbereitet. Kopieren Sie `main.c` aus Verzeichnis `testperf/a0` ins aktuelle Verzeichnis, das Programm können Sie nun mit `make -f makefile.ompp` übersetzen.

3.2.2 Programm ausführen

Setzen Sie zuerst `MAX_NUM` auf (10*1000*1000). Starten Sie das Programm anschliessend mit einer verschiedenen Anzahl von Threads: wählen Sie dazu die Anzahl Threads als Vielfaches der Anzahl CPUs: und zwar mal 1, 2 und 4.

Damit ompP korrekt arbeitet, müssen Sie die Anzahl Threads wählen, indem Sie die Umgebungsvariable `OMP_NUM_THREADS` auf den entsprechenden Wert setzen (Default: Anzahl CPUs auf Ihrem System), zum Beispiel:

```
export OMP_NUM_THREADS=4
```

OmpP erzeugt bei jedem Run einen Profiling Report (Textdatei), der Dateiname enthält dabei die Anzahl gewählter Threads. Weiter Informationen zum Filenamen, etc. finden Sie im User Guide auf Seite 3 unten (Abschnitt 3.2).

3.2.3 Programme analysieren

Analysieren und Interpretieren Sie die Ausgabe in den Reports zum Programm. Informationen dazu finden im User Guide in Abschnitt 4.

Was schliessen Sie aus den Resultaten für das Programm?

3.2.4 Instrumentierung

Schauen Sie sich die von ompP erzeugten Files an: Sie sehen (zumindest in einem gewissen) wie der Programmcode instrumentiert wird.

4 Take Home

Welche Informationen kann ein Profiler zu parallelen Programmen liefern und wie kann diese Information für weitere Optimierungen genutzt werden. Was ist dabei ev. problematisch?

Notieren Sie sich die wichtigsten Punkte, die Sie im Zusammenhang mit diesem Praktikum kennengelernt haben.