

Pflichtenheft - Weiterentwicklung des "Gorynych" Projekts

Marten Wick, Sergej Zuyev 2016

Technische Hochschule Mittelhessen

Inhaltsverzeichnis

Pflichtenheft - Weiterentwicklung des "Gorynych" Projekts.....	1
1. Zielbestimmung.....	3
. 1.1 Musskriterien.....	3
. 1.2 Wunschkriterien.....	3
. 1.3 Abgrenzungskriterien.....	3
2. Produkteinsatz.....	4
. 2.1 Anwendungsbereiche.....	4
. 2.2 Zielgruppen.....	4
. 2.3 Betriebsbedingungen.....	4
3. Produktübersicht.....	5
4. Produktfunktionen.....	6
. 4.1 Grundlagen.....	6
. 4.2 Runtime/Compile-time-Dispatcher.....	6
. 4.3 Basisoperatoren.....	6
. 4.4 Basisfunktionen.....	7
. 4.5 Lineare Algebra.....	8
5. Produktdaten.....	9
6. Technische Produktumgebung.....	9
. 6.1 Betriebssysteme.....	9
. 6.2 Hardware.....	9
. 6.3 Orgware.....	9
7. Qualitätsanforderungen.....	9
8. Benutzeroberfläche.....	9
9. Nicht-Funktionale Anforderungen.....	9
10. Revisionshistorie.....	10
11. Referenzen:.....	10

1. Zielbestimmung

"Gorynych" ist eine templateorientierte Abstraktionsschicht zur Durchführung von beschleunigten Berechnungen auf heterogenen Systemen. Mit diesem Framework ist es möglich, einmalig den Algorithmus zu implementieren und mit Hilfe der automatischen Abbildung von verschiedenen Prozessor-Fähigkeiten zu profitieren.

Ziel der Erweiterung des Projektes ist, die "Write-Once" Philosophie auch für beschleunigte Berechnungen auf hochparallele Grafikprozessoren zu übertragen.

1.1 Musskriterien

Einfachere "Solowej" Module müssen auf dem Grafikprozessor ausgeführt werden. Das sind solche, die ausschließlich Daten verarbeiten oder generieren, aber nicht zwischenspeichern.

1.2 Wunschkriterien

Komplexere "Solowej" Module werden auf dem Grafikprozessor ausgeführt.

1.3 Abgrenzungskriterien

Es ist nicht vorgesehen, zusätzliche Computing-Funktionalitäten einzubringen, die im "gorynych"-Projekt in keinen anderen Berechnungszweigen vorhanden sind. Beispiel: vektorisierte trigonometrische Funktionen. Diese sind im Gesamtprojekt noch nicht unterstützt und werden auch nicht im Rahmen dieses Auftrages betrachtet.

Es ist nicht vorgesehen, das Maximum aus schwächster Hardware herauszuholen. Hauptkriterium ist die Portabilität, die durch dynamische Instruktionsverteilung erreicht wird.

2. Produkteinsatz

2.1 Anwendungsbereiche

"Gorynych" wird in Bereichen verwendet, die keine hochpräzisen Berechnungen benötigen, aber dennoch von der Beschleunigung profitieren sollen.

Zu diesen Bereichen zählen unter Anderem:

- Computerspiele
- Grafische Datenverarbeitung
- Künstliche neuronale Netze

Zu diesen Bereichen zählen **nicht**

- Hochpräzise wissenschaftliche Anwendungen
- Lebenskritische Anwendungen

2.2 Zielgruppen

Die "Gorynych"-Bibliothek zielt auf Softwareentwickler ab, die C++ beherrschen und ihre Algorithmen nach geringfügigen Anpassungen beschleunigt ausführen wollen.

2.3 Betriebsbedingungen

Die "Gorynych"-Bibliothek an sich erfordert kein System, da sie nur eine Sammlung von Hilfsfunktionen und Abstraktionen ist und autark nicht lauffähig ist. Erst ein Produkt, welches "Gorynych" einsetzt, benötigt einen Rechner mit einem x86-kompatiblen Hauptprozessor und optional einen Grafikprozessor mit OpenCL-Unterstützung.

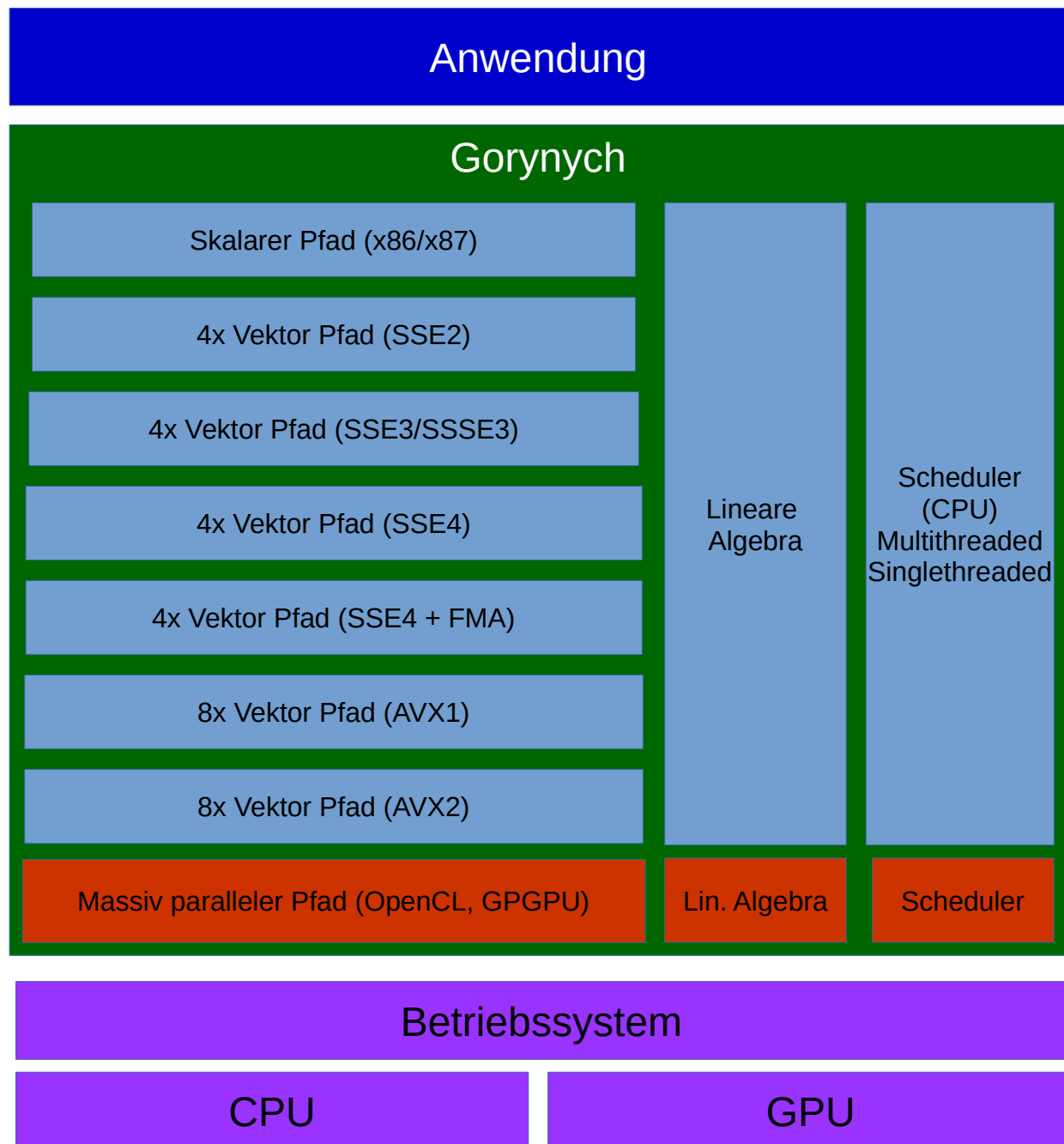
3. Produktübersicht

Grün: „Gorynych“ Abstraktionsschicht

Rot: Innerhalb dieses Auftrags zu implementierende Funktionen

Hellblau: Bereits vorhandene Funktionen

Dunkelblau: Auf „Gorynych“ aufbauendes autarkes Projekt (Nicht im Rahmen des Auftrages)



4. Produktfunktionen

4.1 Grundlagen

- F-1-10 OpenCL-Abstraktion für 32-bit Ganzzahlen
- F-1-20 OpenCL-Abstraktion für 32-bit Gleitkommazahlen
- F-1-30 OpenCL-Scheduler-Basis
- F-1-40 OpenCL-Codegenerierungs Funktionalität
- F-1-50 Plattform-Detektor muss um OpenCL-Detektor erweitert werden

4.2 Runtime/Compile-time-Dispatcher

- F-2-10 Die Typverteiler müssen um den OpenCL-Zweig erweitert werden
- F-2-20 Die Instruktionsverteiler müssen um den OpenCL-Zweig erweitert werden
- F-2-30 Das Build-System muss um den OpenCL-Zweig erweitert werden

Vorbedingung: F-1-X müssen implementiert sein.

Für automatisiertes Testen ist die Verteilung zur Laufzeit nicht von Bedeutung.

4.3 Basisoperatoren

- F-3-10 Addition
- F-3-20 Subtraktion
- F-3-30 Multiplikation
- F-3-40 Division
- F-3-50 Boolesche Operatoren
 - F-3-51 Logisches Und
 - F-3-52 Logisches Oder
 - F-3-53 Logisches Nicht

- F-3-60 Vergleichsoperatoren
 - o F-3-61 Gleichheit
 - o F-3-62 Ungleichheit
 - o F-3-63 Größer-Gleich
 - o F-3-64 Größer
 - o F-3-65 Kleiner
 - o F-3-66 Kleiner-Gleich

Vorbedingung: F-1-10 und F-1-20

4.4 Basisfunktionen

- F-4-10 Rundung zum nächsten Ganzzahlwert
- F-4-20 Rundung zum nächsthöheren Ganzzahlwert
- F-4-30 Rundung zum nächsttieferen Ganzzahlwert
- F-4-40 Abschneiden des Nachkommaanteils
- F-4-50 Absolutwert
- F-4-60 Minimalwert
- F-4-70 Maximalwert
- F-4-80 Selektion

Die o.g Funktionalität gilt sowohl für Vektor- als auch Skalartypen

Vorbedingung: F-1-10, F-1-20, F-5-10.

- F-4-90 Speicherzugriffe
 - o F-4-91 Vektorisierte Werte auslesen
 - o F-4-92 Doppelt indirekte Speicherzugriffe

Vorbedingung: F-1-10 und F-1-20.

4.5 Lineare Algebra

- F-5-10 Vektoren bis $1 \times N$
 - Vorerst sind Dimensionen von 1×2 , 1×3 ausreichend.
- F-5-20 Matrizen bis $N \times M$
 - Vorerst sind Dimensionen von 2×2 , 2×3 , 3×2 , 3×3 ausreichend.
- F-5-30 Grundoperationen
 - F-5-31 Skalar addieren
 - F-5-32 Skalar subtrahieren
 - F-5-33 Skalieren (Multiplikation)
 - F-5-34 Skalieren (Division)
 - F-5-35 Addition
 - F-5-36 Subtraktion
 - F-5-37 Matrixmultiplikation
 - F-5-38 Punktprodukt
 - F-5-39 Kreuzprodukt

5. Produktdaten

Die "Gorynych"-Bibliothek speichert keine Daten, da sie nur eine Sammlung von Abstraktionen und Hilfsfunktionen ist.

6. Technische Produktumgebung

6.1 Betriebssysteme

- Linux
- Windows (8.1, 10)

6.2 Hardware

- x86-kompatibler moderner Hauptprozessor
- OpenCL-kompatibler Grafikprozessor

6.3 Orgware

Nicht vorgesehen.

7. Qualitätsanforderungen

"Gorynych" muss auf gängigen x86-fähigen Prozessoren lauffähig sein, gleiches gilt für Grafikprozessoren mit OpenCL Unterstützung.

8. Benutzeroberfläche

Nicht vorgesehen.

9. Nicht-Funktionale Anforderungen

- NF-1-10 Die Projektarchitektur muss weiterhin modular und wartbar sein.
- NF-1-20 Das Projekt muss erweiterbar bleiben.

10. Revisionshistorie

V1.0 Sergej Zuyev, 13.08.2016

V1.1 Marten Wick, 14.08.2016

V1.2 Sergej Zuyev, 15.08.2016

11. Referenzen:

1. Gorynych: <https://github.com/zz-systems/gorynych>
2. Solowej: <https://github.com/zz-systems/solowej>