

December 5, 2017



PROTHOS

PROGRAMMABLE TASKFLOW ORIENTED
OPERATING SYSTEM

Stefan Wesner, Lutz Schubert, Stefan Bonfert,
Richard Membarth, Randolph Rotta, ...

AGENDA



1. Motivation

2. DSL für Stencil Pipelines

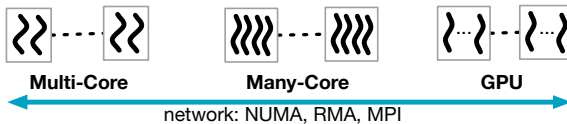
3. DSL für One Dimensional Turbulence

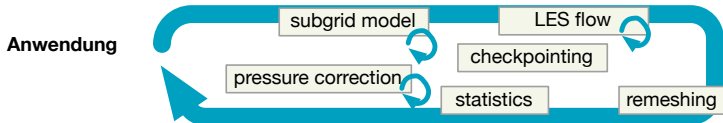
4. Zusammenfassung

Anwendung

Ausführung

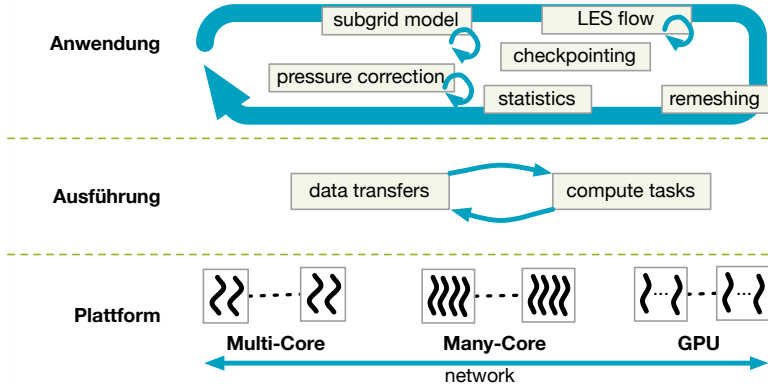
Plattform

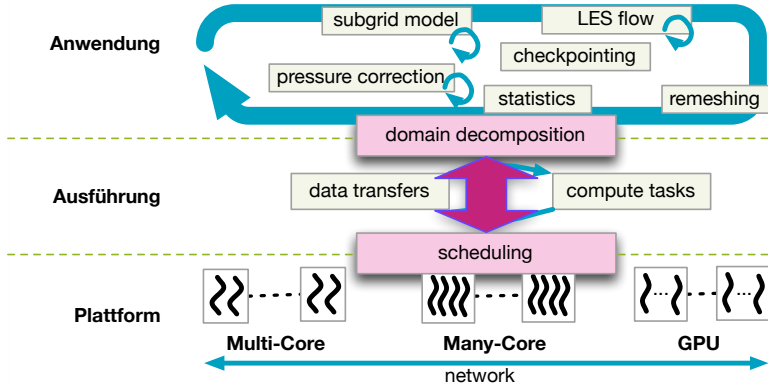


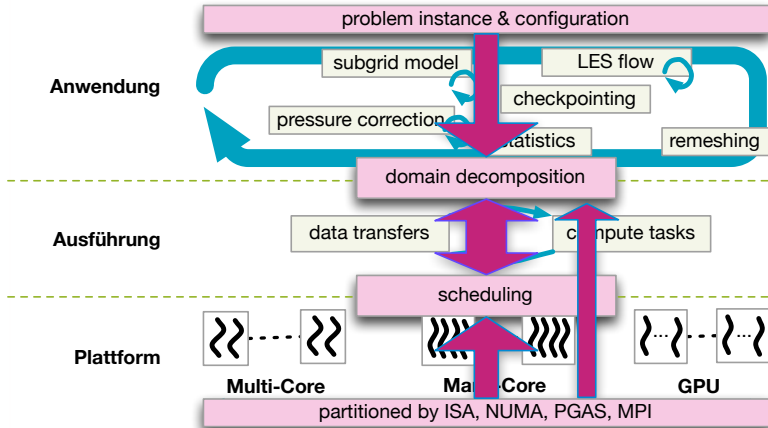


Ausführung









⇒ GERINGE PRODUKTIVITÄT UND PERFORMANCE



- Dekomposition hängt ab von Plattform + Konfiguration + Zustand oft fest in Anwendung (design-time)
- Scheduling Methode ist Kompromiss zw. Task-Größe und Overhead oft fest in Ausführungsumgebung

⇒ Entwickler = Physik- & Numerik- & Parallel-Experte

PERFORMANCE = ABSTRAKTION + TASKIFIZIERUNG

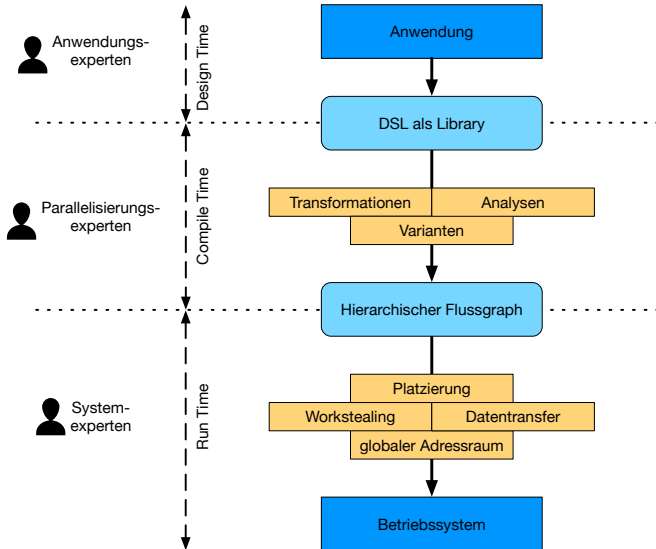


- Daten- und Aufgaben-Parallelismus
- Datenabhängigkeiten identifizieren
- optimale Taskgröße bestimmen
- manuell oder automatisch

State of Art

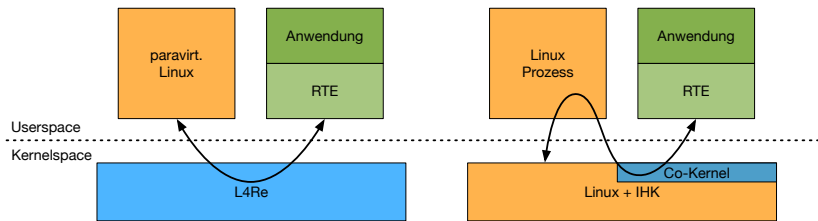
- hoher Overhead des Task- und Daten-Managements (z.B. OmpSs) oder nur eine Shared-Memory Domain (z.B. Intel TBB)

PROTHOS - LÖSUNGSANSATZ



WELCHES BETRIEBSSYSTEM?

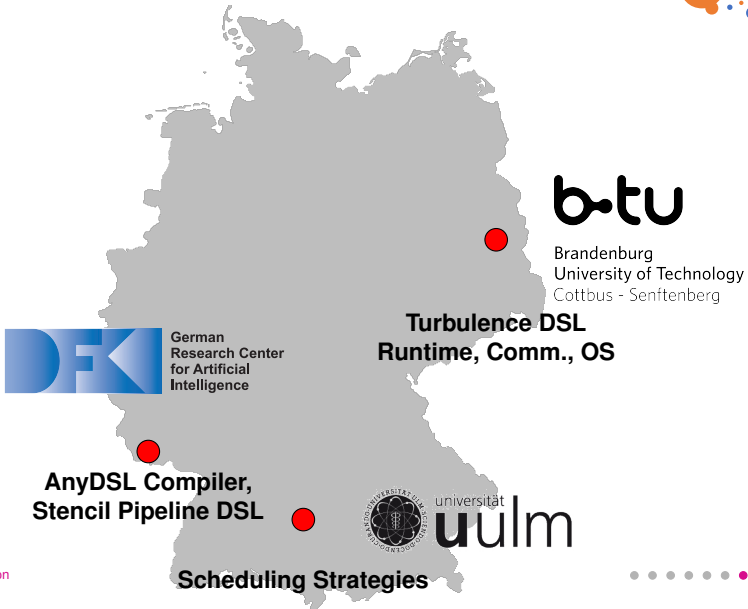
Maßgeschneiderte Runtime Environments für HPC



Co-Kernel

- Hermit Core (RWTH Aachen)
- mOS (Intel)
- mcKernel (Riken)
- MyThOS (in Arbeit)

⇒ Reduzieren Overhead der Thread- und Speicherverwaltung



Abgeschlossen

- Anforderungsanalyse
- Prototyp: Stencil Pipeline
- Prototyp: ODT Code in anyDSL und C++
- Recherche Task- und Datenfluss-Modelle

In Arbeit

- Entwurf der Scheduler-Bausteine
- Integration der Laufzeitumgebung in Linux+MyThOS
- Taskifizierte DSLs für Stencil Pipelines und ODT Codes

AGENDA



1. Motivation

2. DSL für Stencil Pipelines

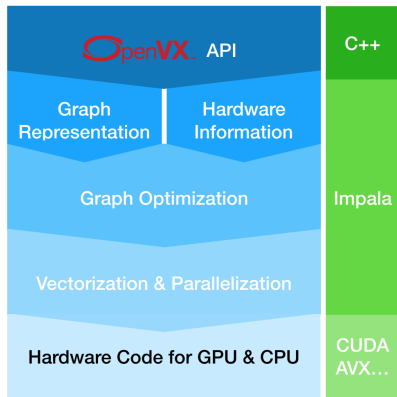
3. DSL für One Dimensional Turbulence

4. Zusammenfassung

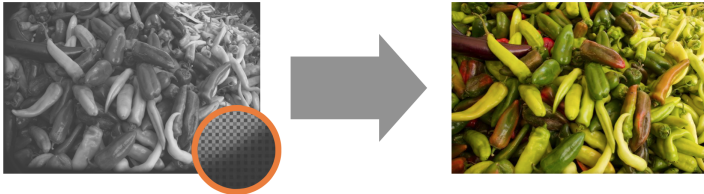
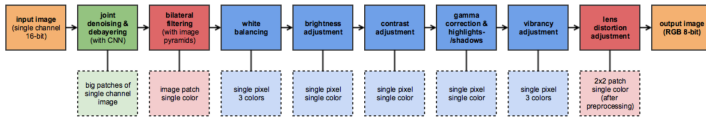
OPENVX IN ANYDSL



- OpenVX (Khronos Group): Graph-basierte, funktionale Beschreibung von Bildverarbeitungsanwendungen
- Implementierung durch Bibliotheken von Hardwareherstellern (Intel, AMD, NVIDIA)
- Übersetzung von OpenVX nach Impala
- Optimierung und Code-Generierung in AnyDSL



BEISPIEL: RAW PIPELINE IN ANYDSL



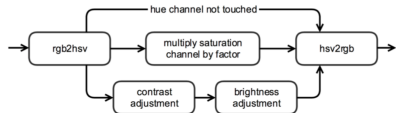
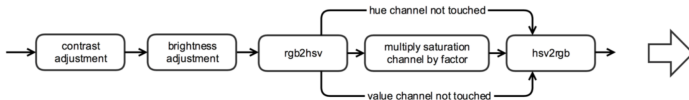
Erste Ergebnisse

- OpenCV: 3× schneller
- Halide: 2× langsamer

DOMÄNENSPEZIFISCHE GRAPH-OPTIMIERUNGEN



- Operationen ohne Einfluss auf Endergebnis löschen
- High-Level Knoten im Graph zerlegen
- Operationen Kombinieren
- Farbraum basierte Optimierung



AGENDA



1. Motivation

2. DSL für Stencil Pipelines

3. DSL für One Dimensional Turbulence

4. Zusammenfassung

ODT-LES

Large Eddy Simulation + stochastic subgrid model



- big eddies (turbulence) solved explicitly, smaller eddies than LES mesh solved with ODT model
- stochastic process: simulate effect of random eddies along 1D line
- dimensional splitting: parallel, independent lines in each dimension

Main Timestep Loop

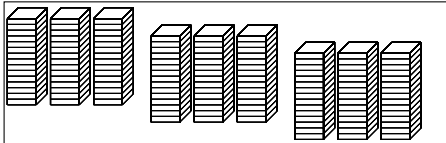
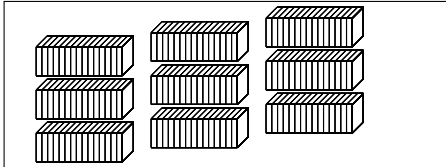
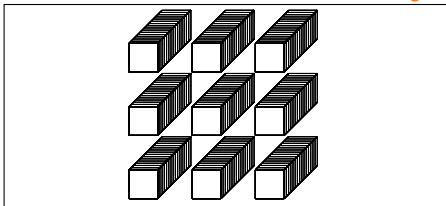
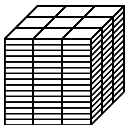
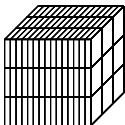
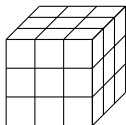
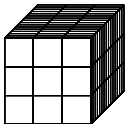
1. Solve Pressure correction (LES)
2. Update velocity values in the X-LES grid
3. Update velocity values in ODT lines
4. Advance ODT lines
5. Correct velocities in X-LES grids
6. Update velocity in LES grid (go back to step 1)
7. +Statistics, +Particle Tracing, +Checkpoints, +Mesh adaption. . .

ODT LINES

3x3x3 LES GRID

3 x (3x3x15 X-LES GRIDS)

3 x (9 x (1x1x15 ODT Lines))



AGENDA



1. Motivation

2. DSL für Stencil Pipelines

3. DSL für One Dimensional Turbulence

4. Zusammenfassung

Ziel: Höhere Produktivität mit guter Performance
für komplexe Anwendungen

Idee: Hierarchische Flow Graphs — Scheduling Nodes,
Compile-Time Optimierung — Partial Evaluation,
Reduzierte Overheads — Spezialisiertes Betriebssystem

