

本スライドは、当日のセミナー資料から一部を抜粋したものです。

GPGPUで応用が広がるGPU

国立大学法人 九州大学
大学院システム情報科学研究所 教授
村上 和彰

GPUとCPUの違い



- コンピュータによる描画に関わる処理
コンピュータによる描画に関わる処理:異なるデータに対し、同じ操作を並列に適用可能
高精細化に従い、より多くのデータに対する操作が必要に

高性能化:より多くの演算器を搭載



描画以外の処理でも利用

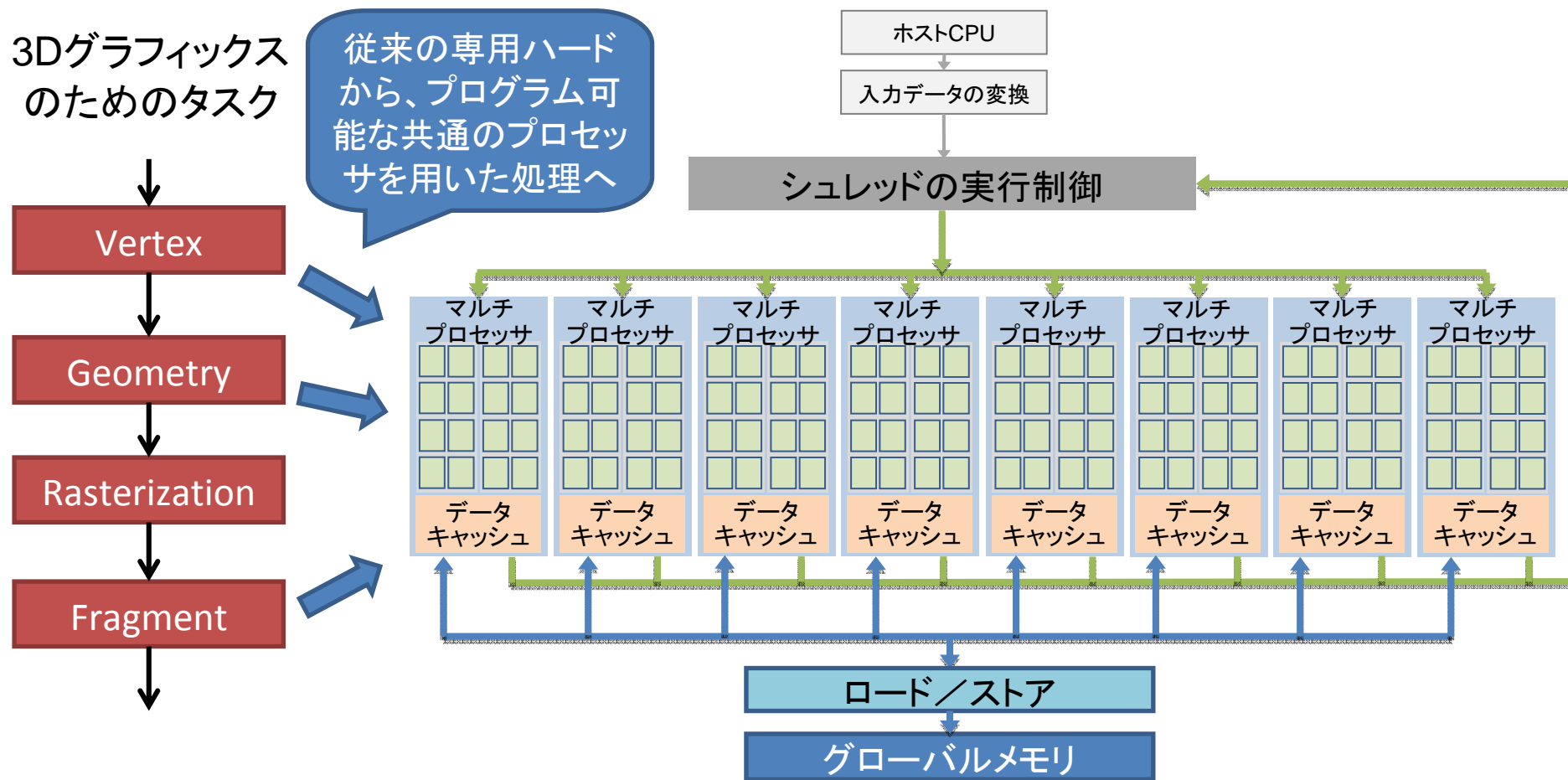


- コンピュータの様々な用途(アプリケーション)に対応
- 高性能化のために、様々な手法を利用
 - メモリレイテンシの隠蔽
 - 命令、シュレッドレベルの並列化、実行予測機構

高性能化:キャッシュメモリ、並列化のための制御機構、高MHz化、マルチコア化

GPUのアーキテクチャの例

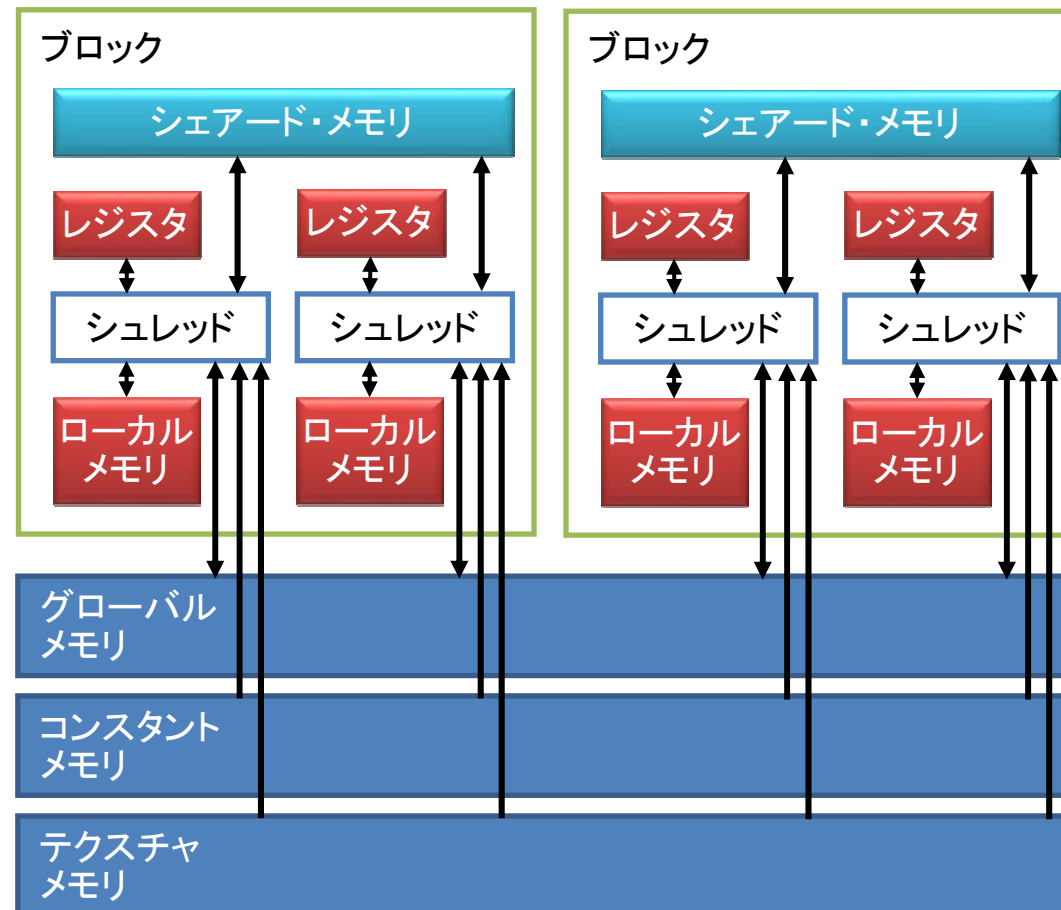
～ GeForce 8800 ～



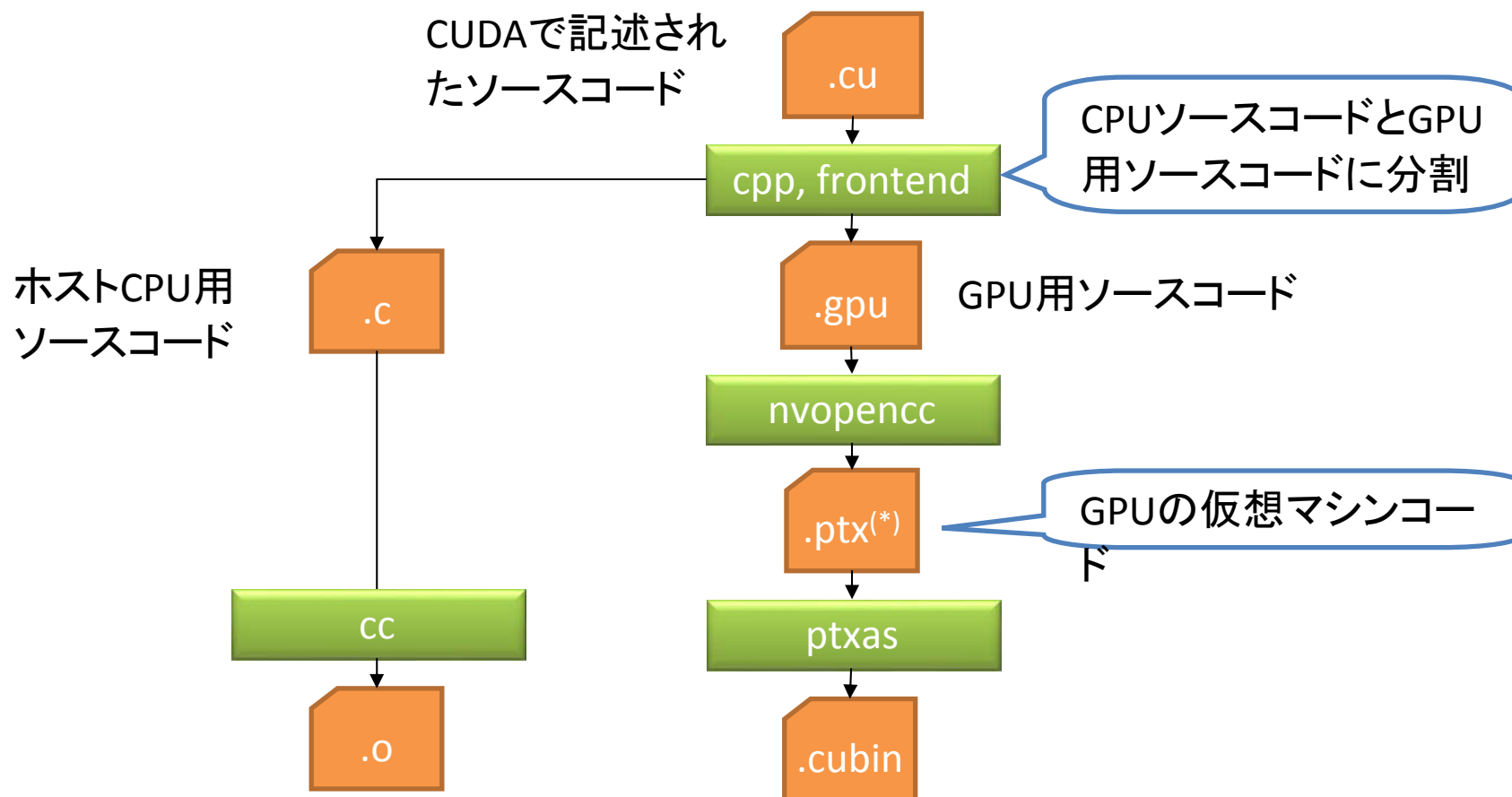
GPUを用いたアプリケーションの実行 ～CUDAを用いた例～

実行の単位(シュレッド、ブロック、グリッド)により、利用可能なメモリが異なる
•アプリで、利用可能なメモリのサイズ、シュレッド間でのデータの受け渡し等の考慮が必要

- CUDAのメモリモデル



オブジェクトコードの生成 ～NVIDIA社のコンパイラの例～



ホスト(CPU)用、GPU用のそれぞれのオブジェクトコードを出力

GPGPUのための主な開発環境

- GPGPUのソフトウェアのための規格

	開発の主導	GPGPUへの対応方法	対応するGPU
CUDA	NVIDIA	C/C++の拡張、API(*) (関数) の定義	NVIDIA GPU
OpenCL	Khronosグループ (業界コンソーシアム)	C/C++の拡張、API(*) (関数) の定義	NVIDIA, Intel, AMD, Apple社、ARMを始めとする複数ベンダのGP
DirectCompute	マイクロソフト	API(*) (関数) の定義	DirectX11 (グラフィックス、マルチメディア用API) が利用可能なGPU

- その他プログラム言語での対応
 - Fortran, Java, Python

(*) API: Application Program Interface