Porting Scientific codes to GPUs

- Common characteristics of scientific codes:
  - Usually in Fortran
  - FFT-based simulations involve all-to-all communication
  - High memory requirement
- Incompatibility with GPUs:
  - GPUs have small on-chip memory (1-16GB max)
  - Communication latencies in data movement

Solutions for porting code:
- Domain decomposition (regular or irregular)
- Explicit properties of data and communication kernel
- Sampling/pruning used so that domain results fit on GPU memory

Combining performance with scaling scientific codes requires algorithm restructuring.

Case study: MASSIF
- Hybrid saxo simulation
- Partial Differential Equation solved by Green’s function
- FFT-based convolution and tensor contraction between rank-2 tensors and rank-4 Green’s function

Proposed algorithmic solution:
- Domain decomposition with grains are domains
- Domain-local FFT followed by convolution and tensor contraction
- Green’s function computed on-the-fly to avoid storage
- Adaptive sampling of dense convolution result to fit problem on GPU memory

Complex data mappings! How to get maximum performance on various platforms?

OGX-2
Source: NVIDIA
Summit Source: ORNL

Challenges
- FFTW is de-facto standard interface for FFT
- Vander libriaseship the FFTW 3.x interface: Intel MKL, IBM ESSL, AMD ACML, Netlib (of-kinds), Nvida cuFFT, Cray LibSciCRAFT
  - Some issues:
    - No native support for accelerators (GPUs, Xeon Phi, FPGAs) and SMIs
    - Parallelization function does not scale beyond 32 nodes
    - No analogue to LAPACK for spectral method

Solution: Emerging interfaces like FFTX, extension of FFTW, enables algorithm specification as composition of sub-plans

Front end: Algorithm Specification

FFT, tensor contraction and sampling

Output data size

Octree Sampling reduces storage on GPU

Back end: Code Optimization

FFTX backend: SPIRAL

Platform-aware formal program synthesis

Translating an OL expression into code

Constraint Solver Input:

Output = Constraint expression

C code: (FFTW plan copied over all domains)

Model: common abstractions = spaces of matching formulas

Automatically generated FFTW-like library components

DARPA BRAINS

Future Plans + Other Applications

Future work: MASSIF
- Irregular domain decomposition
- Extension of adaptive sampling for irregular domains

Future work: FFF and SpectralPACK
- Numerical Linear Algebra
- Spectral Algorithms

Poisson’s equation in free space

Partial differential equation (PDE) Solution

Hockney free-space convolution

Hockney: Convolution + problem-specific zero padding and output subset

2012 HPC Challenge Class C Award (Most Productive System) with ANL and IBM

References

Dr. A. Kulkarni, Franchetti, and J. Kova

Octree-based sampling pattern is encoded by a descriptor

CODING is a plan: axes are studied.

In: A2 (Data plan) where A2 = [ (m,n), k]...i = (m,n)

other FFT

LAPACK for spectral algorithms

- Define FFT as the analogues to BLAS
- Define class of numerical algorithms to be supported by SpectraPACK
- POD solver classes (circular convolution, sparse in normspace, ...), signal processing
- Define SpectraPACK functions (circular convolutions, NUFFT, Poisson solvers, free space convolution

References