Lecture 7 - Introduction to Performance Optimization

DSE 512

Drew Schmidt
2022-02-15

From Last Time

- New Homework
- Questions?

Where We've Been

Module 1: Basic Cloud and HPC

- Lecture 1 - Introduction
- Lecture 2 - Overview of HPC and the Cloud
- Lecture 3 - Introduction to Remote Computing
- Lecture 4 - Introduction to Containers
- Lecture 5 - Introduction to ISAAC
- Lecture 6 - MPI and Singularity

Where We're Headed

Module 2: Performance Optimization

- High Level Language Optimizations
- I/O
- Computational Linear Algebra
- GPGPU: The Easy Parts
- Utilizing Compiled Code

Where's the Data Science?

So Your Software Is Slow

- Is it actually slow?
- What does that even mean?
- Do you have an I/O problem? A compute problem? Memory?
- Is it a HLL (R/Python)
  - Using vectorization?
  - Using efficient kernels?
  - Can you rewrite it in C?
- Is it linear algebra dominant?
  - Are you using fast BLAS?
  - Are you using multi-threaded BLAS?
- Can it be parallelized?

High Level Language Optimizations

- General strategies apply
- Implementation(s) very language dependent
- Examples in R and Python

Optimizations

HLL Strategies

- Compilation concerns
- Use efficient kernels/packages
- Vectorization
- JIT and/or bytecode compilers
- Fundamental types
- Language quirks (e.g. `if` vs `ifelse` cost in R)

Other Concerns

- I/O
- Linear algebra libraries
- Advanced hardware, e.g. GPGPU
- Utilizing compmiled code
- Parallelism

I/O

- Different strategies
  - plain text
  - binary
  - database
- Serial vs Parallel
  - Serial hard to get wrong
  - Parallel hard to get right
  - lustre vs HDFS

Recall: Terminology

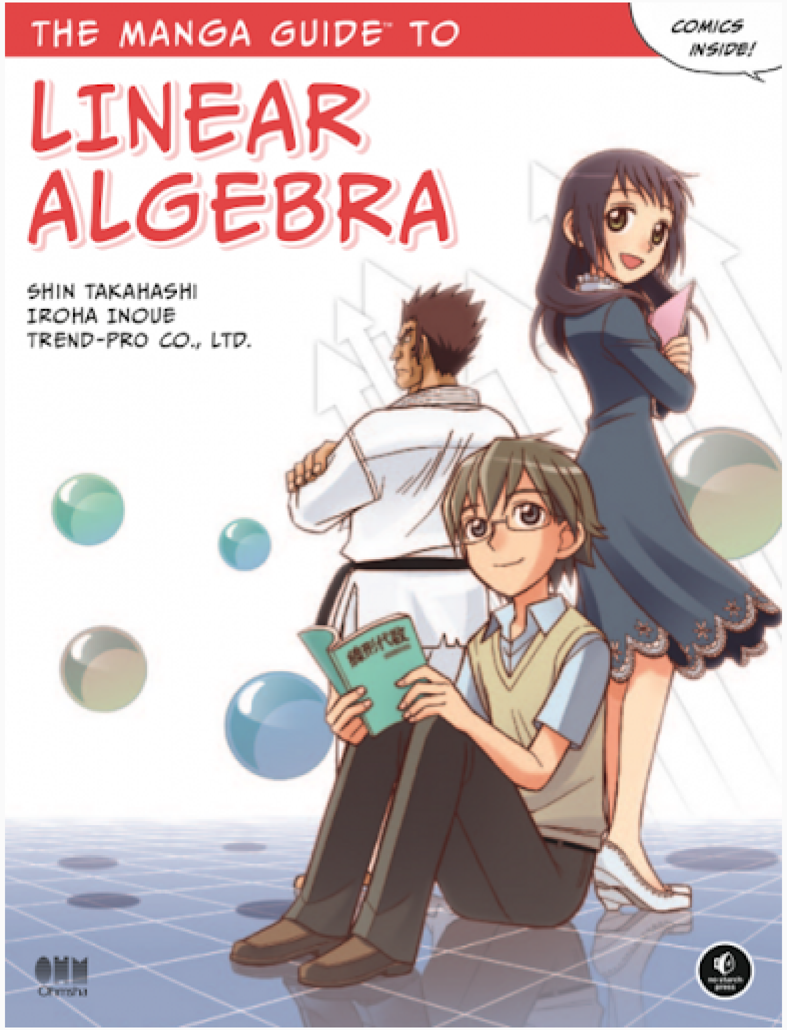
- **gemm** - matrix-matrix multiply
- **BLAS** - Basic Linear Algebra Subprograms; matrix library
- **FLOPS** - Floating Point Operations Per Second (adds and multiplies)
- **LINPACK** - Solve $Ax = b$
- **TOP500** - list of computers ranked by LINPACK benchmark

The LINPACK Benchmark

- Solve the system $Ax = b$
  - A- $n \times n$ matrix (you choose n)
  - Double precision
  - Must use LU with partial pivoting
    - $A = LU$
    - $b = Ax = LUx$
- $\frac{2}{3}n^3 + 2n^2$ operations
- Solution must satisfy some accuracy conditions.
- Most FLOPS wins!

Linear Algebra

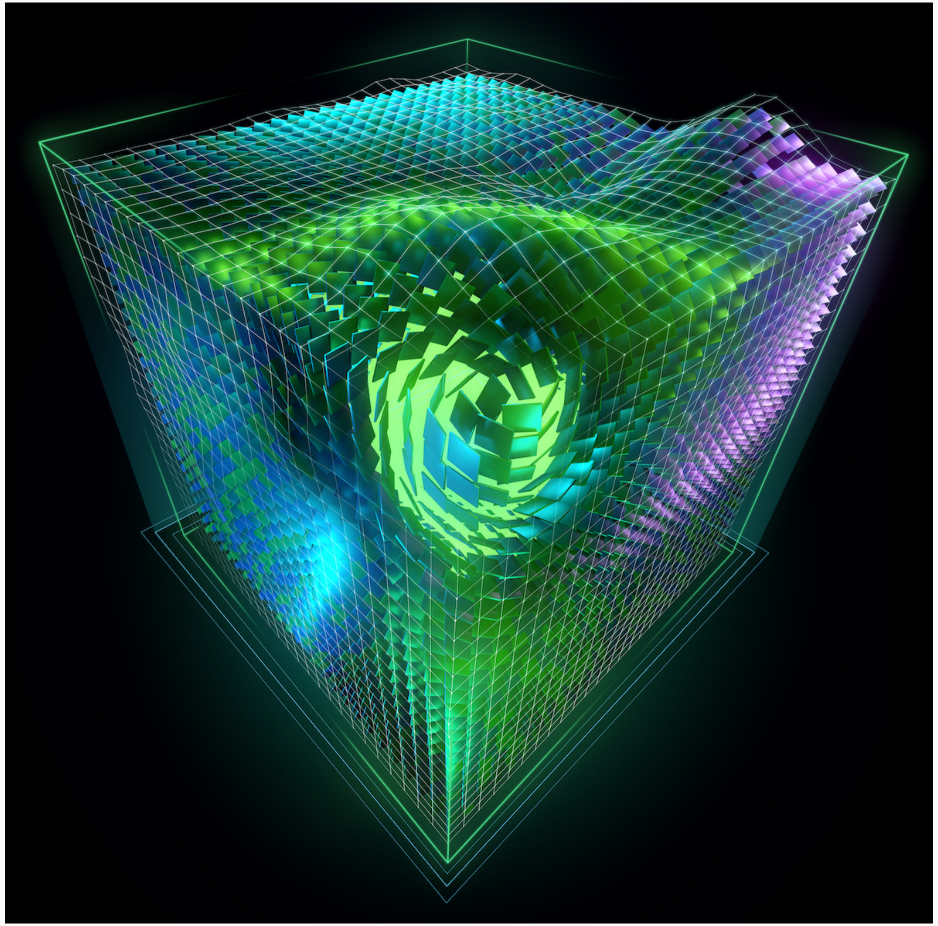
- LA dominates scientific and data computing
- Some uses in data:
  - PCA - SVD
  - Linear Models - QR
  - Covariance/correlation - gemm/syrk
  - Inverse - Cholesky, LU
- 1970's: LINPACK (not that one)
- 1980's: BLAS, LAPACK
- 1990's: ScaLAPACK
- 2000's: PLASMA, MAGMA
- 2010's: ~~DPLASMA~~ SLATE



- Using Video Game Hardware to Multiply Matrices
- Major players
  - NVIDIA
  - AMD
  - Intel…?!?!
- Pros:
  - Fast
  - When you give up, you can mine bitcoin Cons:
  - Hard to program
  - Expensive

"Low-Level" GPGPU Technologies

- ~~Shaders~~
- CUDA
- OpenCL
- OpenACC
- OpenMP

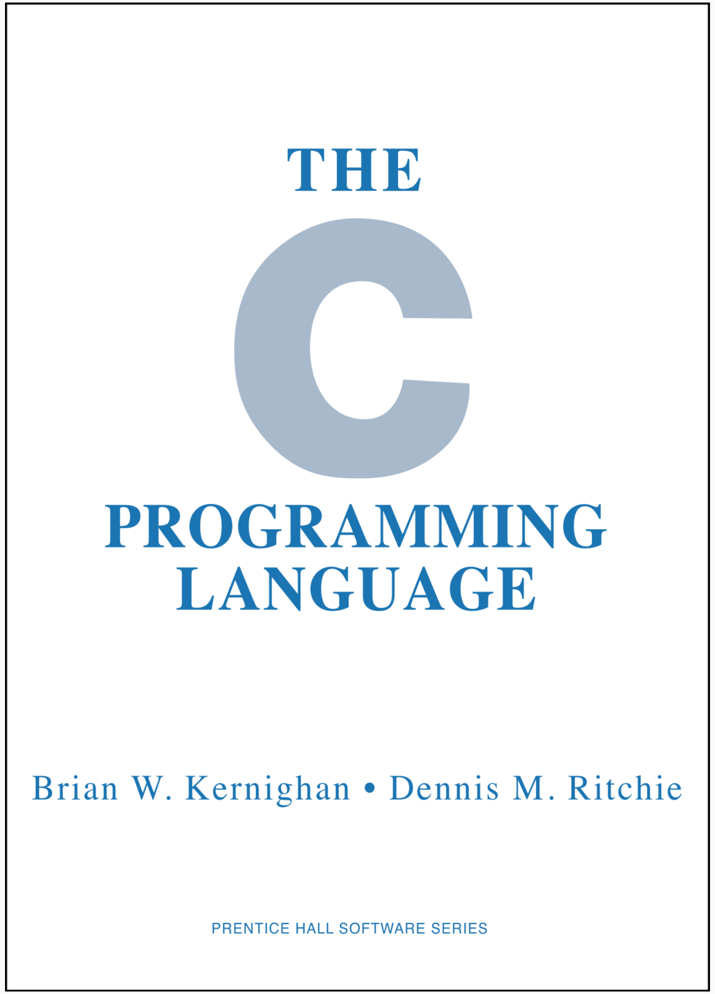


"High-Level" GPGPU Technologies

- Python
    - CuPy
- R
    - fmlr
    - gpuR
- Deep Learning frameworks

Using Compiled Code in a HLL

- Pros
  - fast
  - memory-efficient
  - best of both worlds
- Cons
  - hard to write
  - hard to debug
  - multiple skillsets
- Julia???



Questions?