The Scientific Computing Stack

Nick Torenvliet[1][2]

[1]Systems Design Engineering, University of Waterloo ntorenvl@uwaterloo.ca
[2]Specialized Tool Engineering, Bruce Power nick.torenvliet@brucepower.com

The Scientific Computing Stack

Covered today:

- Hardware
- Operating systems
- Software languages and libraries
- Development tools and workflows
- Working with your IT department
- Tour a working desktop

The Scientific Computing Stack: Your IT Department

IT departments are utilitarian in that they are doing their job when they do the greatest good for the greatest number of people. Meeting your requirements will often be a challenge for them.

The Scientific Computing Stack

- Apologies if your favorite XX.YY V3.14 is not mentioned
- Too many options to provide full coverage
- Within scope of this discussion are many subjective definitions and possible interpretations. Apologies if the current set of definitions and interpretations stray from your version.

The Scientific Computing Stack

Q:What is scientific computing?

A: Today we'll view scientific computing as that which involves the application of advanced methods from mathematics, statistics, and computer science to perform calculations in a nuclear context.

Examples: Monte Carlo methods as applied in nuclear safety analysis (fuel safety analysis), statistical methods as applied in time series analysis (DCC data anomaly detetction and pump vibration monitoring), or machine learning as applied in the analysis of steam generator or fuel channel inspection data.

The Scientific Computing Stack

Progress in scientific computing is highly correlated with available computing power, and sophistication of methods.

- The number of beans our computing devices can count
- The speed at which our computing devices can count those beans
- The efficiency of our bean counting strategies
- Methods influence hardware design and vice versa

The Scientific Computing Stack

The full stack:

- Hardware: Processors, RAM, Disk, GPU
- Operating System: Linux, MacOS, Windows
- Software languages and libraries: Python, R, Julia
- Software libraries: scipy, sklearn, pytorch, tensorflow, jax
- Development tools: VSCode, Git, Jupyter Notebook/Lab

The Scientific Computing Stack

Hardware reads, writes, transports, and operates on data:

- Processors
- Motherboards/Busses
- RAM
- Disk
- GPU
- Networking
- Cloud

Access times, throughput rates, clock frequency, flops

The Scientific Computing Stack

Scientific computing is about passing around and processing data quickly.

- Bytes:
  - where do they reside
  - how often do you read them
  - how often do you write them
  - what operations need doing
  - what is serial and what is parallel

The Scientific Computing Stack

Identify and speed up up your bottlenecks.

Bottlenecks vary between local and cloud based systems.

Bottlenecks vary between development and production systems.

The Scientific Computing Stack: Hardware

Clock frequency, flops, cache, cores, bit-width

- Processor parameters
  - Clock rate
  - Core count
  - Cache
- Processor manufacturers
  - Intel
  - AMD
  - Apple

The Scientific Computing Stack: Hardware

Read/write speeds, cache, and throughput

- RAM
- Disks
  - SSD
  - Platter
- Data Busses
- Networks

The Scientific Computing Stack: Hardware

Graphical Processing Unit - GPU

- GPU Parameters
  - Clock rate
  - RAM
  - GPU core count
- GPU manufacturers
  - NVIDIA - CUDA drivers
  - Google (TPU)

Other video cards do exist, but are not well-supported

The Scientific Computing Stack: OS

Operating systems allow the software to access the hardware:

- Linux
- MacOS
- Windows

The Scientific Computing Stack: OS

Operating systems allow the software to access the hardware:

- Linux
  - Universally well-supported... mostly
  - A computer scientist's OS
  - Ubuntu, Redhat, Debian
  - community driven

Linux is often found in scientific computing environments and IT backend infrastructure.

Many vendors sell linux friendly configurations.

The Scientific Computing Stack: OS

Operating systems allow the software to access the hardware:

- MacOS
  - Well supported for most use-cases
  - Good security
  - Especially well suited for media and content development
  - Deep learning on GPU is challenging

The Scientific Computing Stack: OS

Operating systems allow the software to access the hardware:

- Windows
  - Good desktop experience for average business user
  - Many commercial sol'ns: Matlab, Labview, Visual Studio
  - IT Department friendly
  - Often used as a front end to web or cloud based solutions and deployments

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Numerical and Symbolic Methods
- Heuristic rules based AI
- Probabilistic programming
- Kernel Methods
- Supervised and unsupervised learning
- Reinforcement learning
- Discriminative and generative learning
- GPU Hardware/resource access

These methods comprise innovation.

The Scientific Computing Stack: Languages

Languages with libraries supporting the methods...

- Python
  - the well established new comer
- R
  - the veteran statistician's choice
- Julia
  - the newcomer
- C, C++, Java, Fortran, Pascal, Cobol, Go, Haskell, many more

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Numerical and Symbolic Methods
- https://en.wikipedia.org/wiki/Numerical_analysis
- https://en.wikipedia.org/wiki/Symbolic_method
- Numerical methods:Python/numpy https://numpy.org/
- Symobolic methods:Sage https://doc.sagemath.org/

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Heuristic rules based AI
- https://en.wikipedia.org/wiki/Rule-based_system
- Python/scikit-learn https://scikit-learn.org/

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Probabilistic programming
- https://en.wikipedia.org/wiki/Probabilistic_programming
- PyMC/Python https://www.pymc.io/welcome.html
- turing.jl/Julia https://turing.ml/stable/

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Kernel Methods
- https://en.wikipedia.org/wiki/Kernel_method
- Python/scikit-learn https://scikit-learn.org/

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Supervised and unsupervised learning
- https://en.wikipedia.org/wiki/Supervised_learning
- https://en.wikipedia.org/wiki/Unsupervised_learning

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Reinforcement learning
- https://en.wikipedia.org/wiki/Reinforcement_learning

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- Generative learning $P(x, y)$
- Discriminative learnining $P(x|y)$
- https://en.wikipedia.org/wiki/Generative_model

The Scientific Computing Stack

Languages and libraries allow you to implement methods:

- GPU Hardware/resource access
- Compute Unified Device Architectue - CUDA
- https://developer.nvidia.com/developer-program

The Scientific Computing Stack: Libraries

Python libraries that manage machine learning...

- Tensorflow
  - Google product
  - Large deployments
- Pytorch and Pytorch Lightning
  - Facebook product, now open source
  - Academic
  - Desktop friendly
- JAX
- Keras, Cafe, Theano... others

Allows your code to access your GPU, and provides implementations of the various methods. Work towards parallelization and cluster super computing is on going.

The Scientific Computing Stack: Libraries

- Graphing
- GUI
- Database access
- Signal Processing
- Image Processing
- High efficiency file formats
- Proprietary interfaces
- Low level language interfaces
- Much more

THERE IS a Python library for that...

The Scientific Computing Stack: Development Tools

- CLI - command line interface
- IDE - monolithic integrated development environment
- Editors - less integrated development environment
- Code Versioning

The Scientific Computing Stack: Desktop Workflow

Linux/UNIX guru circa 1970-2000:

- Unix/Linux
- CLI
- VI / EMACS
- plugins

Richard Stallman, Linus Torvalds, coders working on SORO

The Scientific Computing Stack: Desktop Workflow

Corporate development:

- Windows
- Cloud based Resources
- Monolithic IDE
- Modern editor like VSCode

The Scientific Computing Stack: Desktop Workflow

Typcial researcher and development desktop:

- Linux - Ubuntu
- VSCode and associated plugins
- Python and any required supports
- Pytorch Lightning
- Jupyter Notebook/Lab
- git
- Network/Cloud/Local

The Scientific Computing Stack: Your IT Department

IT departments are utilitarian, in that they are doing their job when they do the greatest good for the greatest number of people. Meeting the unique requirement requirements of scientific computing users may be a challenge for them.

- Approved vendors
- Support of non standard hardware and configurations
- Cyber security
- Data and nuclear export restrictions
- Production

The Scientific Computing Stack: Your IT Department

IT departments are utilitarian, in that they are doing their job when they do the greatest good for the greatest number of people. Meeting the unique requirement requirements of scientific computing users may be a challenge for them.

- Approved vendors

The Scientific Computing Stack: Your IT Department

IT departments are utilitarian, in that they are doing their job when they do the greatest good for the greatest number of people. Meeting the unique requirement requirements of scientific computing users may be a challenge for them.

- Support of non standard hardware and configurations

The Scientific Computing Stack: Your IT Department

IT departments are utilitarian, in that they are doing their job when they do the greatest good for the greatest number of people. Meeting the unique requirement requirements of scientific computing users may be a challenge for them.

- Cyber security

The Scientific Computing Stack: Your IT Department

IT departments are utilitarian, in that they are doing their job when they do the greatest good for the greatest number of people. Meeting the unique requirement requirements of scientific computing users may be a challenge for them.

- Data and nuclear export restrictions

The Scientific Computing Stack: Your IT Department

IT departments are utilitarian, in that they are doing their job when they do the greatest good for the greatest number of people. Meeting the unique requirement requirements of scientific computing users may be a challenge for them.

- Production

The Scientific Computing Stack: Quick Tour

- Desktop tour

Discussion

Questions?