UC San Diego

DSC 102
Systems for Scalable Analytics

Spring 2024

Haojian Jin

Now for the course structure …

DSC 102 will get you thinking about the fundamentals of systems for scalable analytics

1. "**Systems**": What resources does a computer have? How to store and efficiently compute over large data? What is cloud?
2. "**Scalability**": How to scale and parallelize data-intensive computations?
3. **For "Analytics"**:
    1. **Source**: Data acquisition & preparation for ML
    2. **Build**: Model selection & deep learning systems
    3. **Deploying** ML models
4. Hands-on experience with scalable analytics tools

Data Systems Concerns in ML

Key concerns in ML:

Accuracy

Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:

Scalability (and efficiency at scale)

Usability

Manageability

Developability

Q: How do "ML Systems" relate to ML?

ML Systems : ML :: Computer Systems : TCS

*Long-standing concerns in the **DB systems** world!*

Q: [illegible] ML systems?

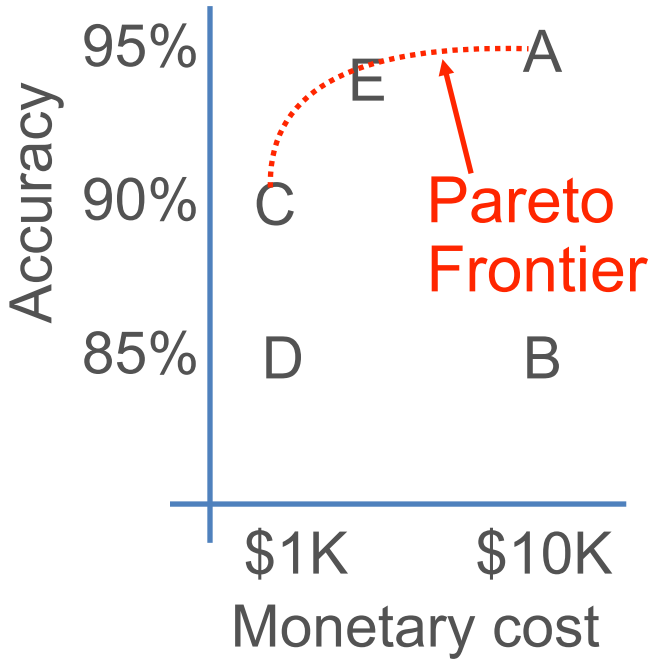
Conceptual System Stack Analogy

	Relational DB Systems	ML Systems
Theory	First-Order Logic Complexity Theory	Learning Theory Optimization Theory
Program Formalism	Relational Algebra	Tensor Algebra Gradient Descent
Program Specification	SQL	TensorFlow? Scikit-learn?
Program Modification	Query Optimization	???
Execution Primitives	Parallel Relational Operator Dataflows	Depends on ML Algorithm
Hardware	CPU, GPU, FPGA, NVM, RDMA, etc.	

Real-World ML: Pareto Surfaces

Q: *Suppose you are given ad click-through prediction models A, B, C, and D with accuracies of 95%, 85%, 90%, and 85%, respectively. Which one will you pick?*

Q: *What about now?*



- Real-world ML users must grapple with multi-dimensional *Pareto surfaces*: accuracy, monetary cost, training time, scalability, inference latency, tool availability, interpretability, fairness, etc.
- *Multi-objective optimization* criteria set by application needs / business policies.

Learning Outcomes of this course

- **Explain** the basic principles of the memory hierarchy, parallelism paradigms, scalable data systems, and cloud computing.
- **Identify** the abstract data access patterns of, and opportunities for parallelism and efficiency gains in, data processing and ML algorithms at scale.
- **Outline** how to use cluster and cloud services, dataflow ("Big Data") programming with MapReduce and Spark, and ML tools at scale.
- **Apply** the above programming skills to create end-to-end pipelines for data preparation, feature engineering, and model selection on large-scale datasets.
- **Reason** critically about practical tradeoffs between accuracy, runtimes, scalability, usability, and total cost.

Tentative Course Schedule

Week	Topic
	Basics of Machine Resources: Computer Organization
	Basics of Machine Resources: Operating Systems
4	Basics of Cloud Computing
4-5	Parallel and Scalable Data Processing: Parallelism Basics
	Midterm Exam on TBD
6-7	Parallel and Scalable Data Processing: Scalable Data Access
7-8	Parallel and Scalable Data Processing: Data Parallelism
9	Dataflow Systems
10	ML Model Building Systems
11	**Final Exam on Dec 15**

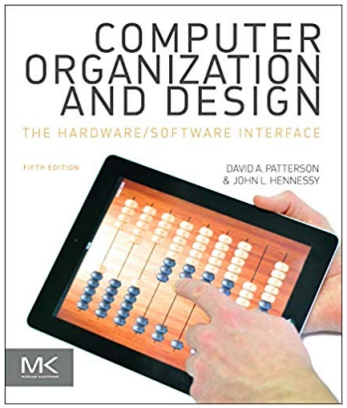
Systems Principles

Scalability Principles

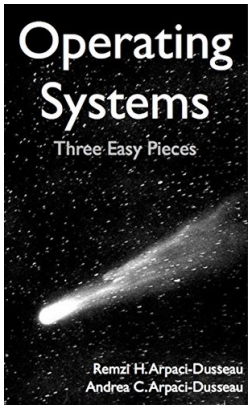
Scalable Analytics Systems

There will be 2 industry guest lectures (maybe 3)

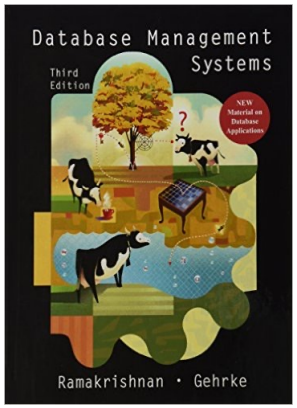
Suggested Textbooks



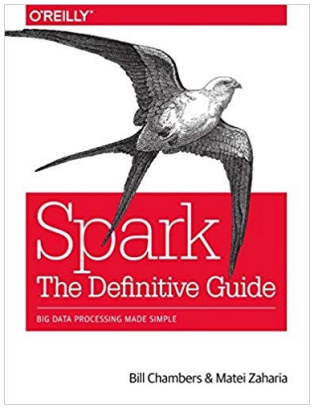
Aka “CompOrg Book”



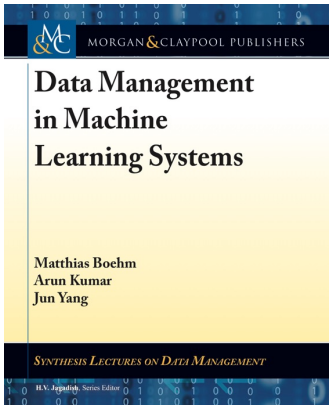
Aka “Comet Book”



Aka “Cow Book”



Aka “Spark Book”

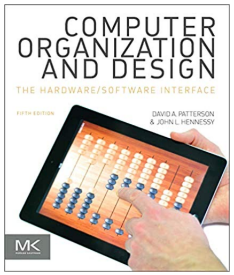


Aka “MLSys Book”

(Free PDFs available online; also check out our library)

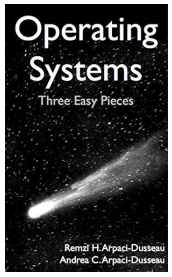
Why so many textbooks?!

1. Computer systems are about carefully layering *levels of abstraction*.

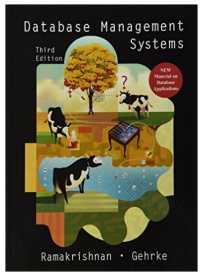


Hardware

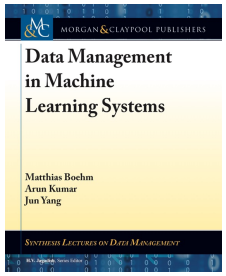
Low-level systems software

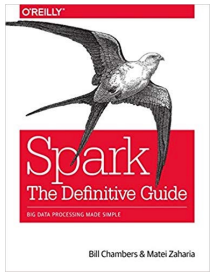


Higher-level relational dataflows



ML-oriented dataflows & lifecycle





More general scalable dataflows

2. Analytics/ML Systems is a recent/emerging area of research.
3. Also, DSC 102 is the first UG course of its kind in the world!

Tentative Course Schedule

Week	Topic
1-2	Basics of Machine Resources: Computer Organization
	Basics of Machine Resources: Operating Systems
	Basics of Cloud Computing
4-5	Parallel and Scalable Data Processing: Parallelism Basics
6	**Midterm Exam on TBD**
6-7	Parallel and Scalable Data Processing: Scalable Data Access
7-8	Parallel and Scalable Data Processing: Data Parallelism
9	Dataflow Systems
10	ML Model Building Systems
11	**Final Exam on Dec 15**

Systems Principles

There will be 2 industry guest lectures (maybe 3)

UC San Diego

DSC 102
Systems for Scalable Analytics

Topic 1: Basics of Machine Resources

Part 1: Computer Organization

Ch. 1, 2.1-2.3, 2.12, 4.1, and 5.1-5.5 of CompOrg Book

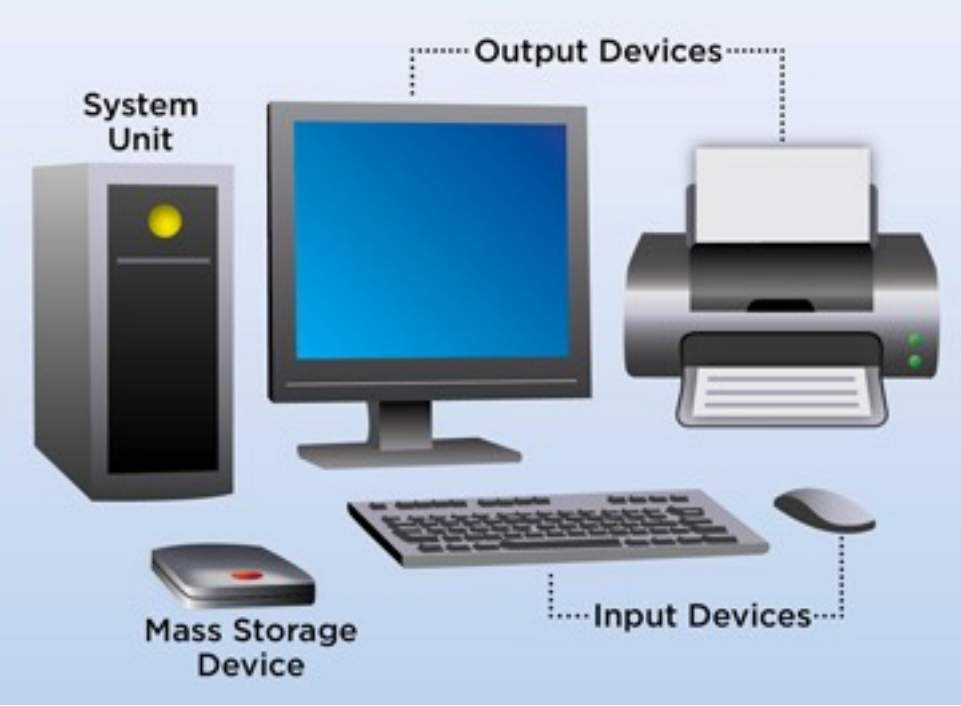
Q: *What is a computer?*

A programmable electronic device that can store, retrieve, and process digital data.

Outline

- Basics of Computer Organization
  - Digital Representation of Data
  - Processors and Memory Hierarchy
- Basics of Operating Systems
  - Process Management: Virtualization; Concurrency
  - Filesystem and Data Files
  - Main Memory Management
- Persistent Data Storage

Parts of a Computer



Hardware:
The electronic machinery (wires, circuits, transistors, capacitors, devices, etc.)

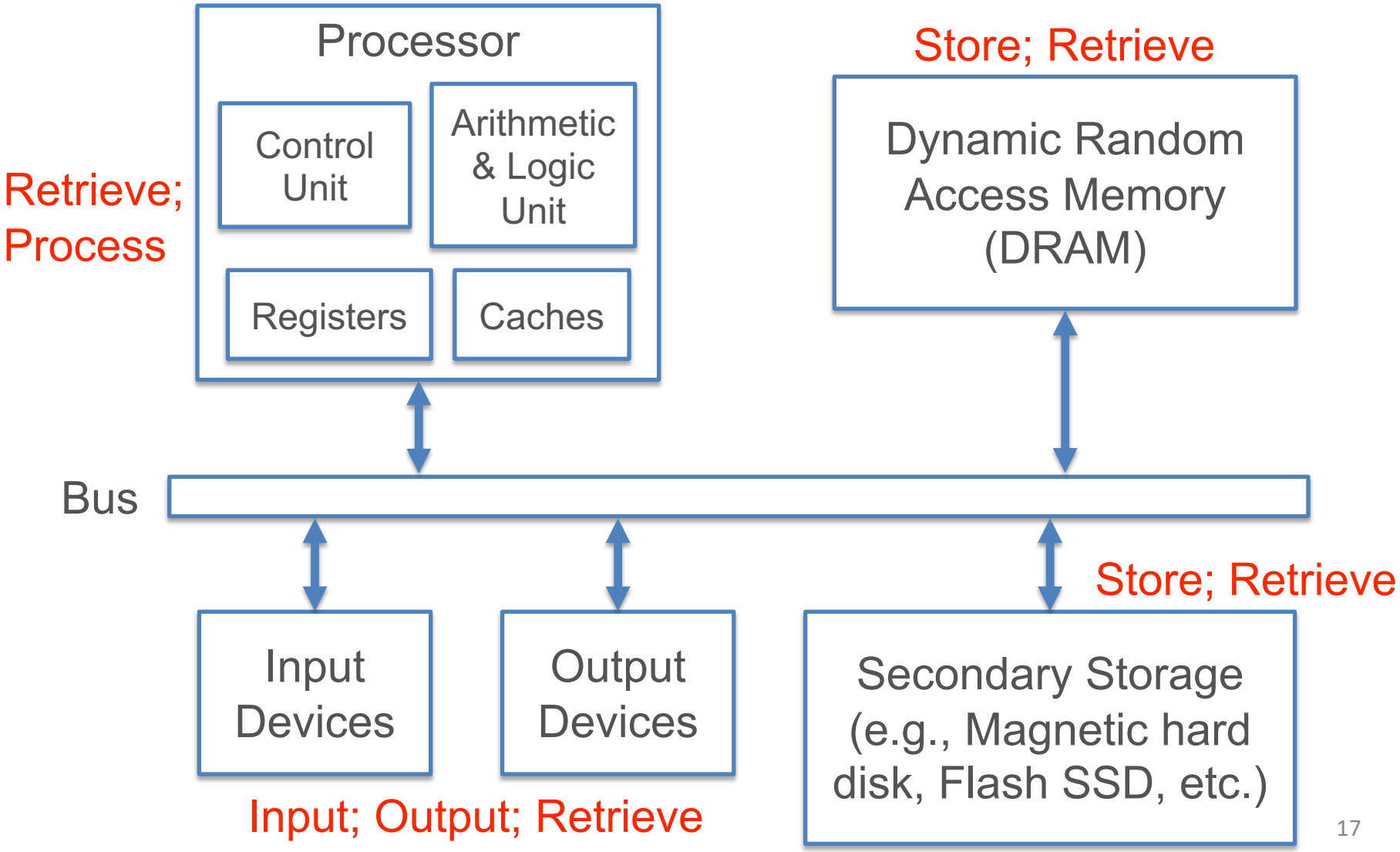
Software:
Programs (instructions) and data

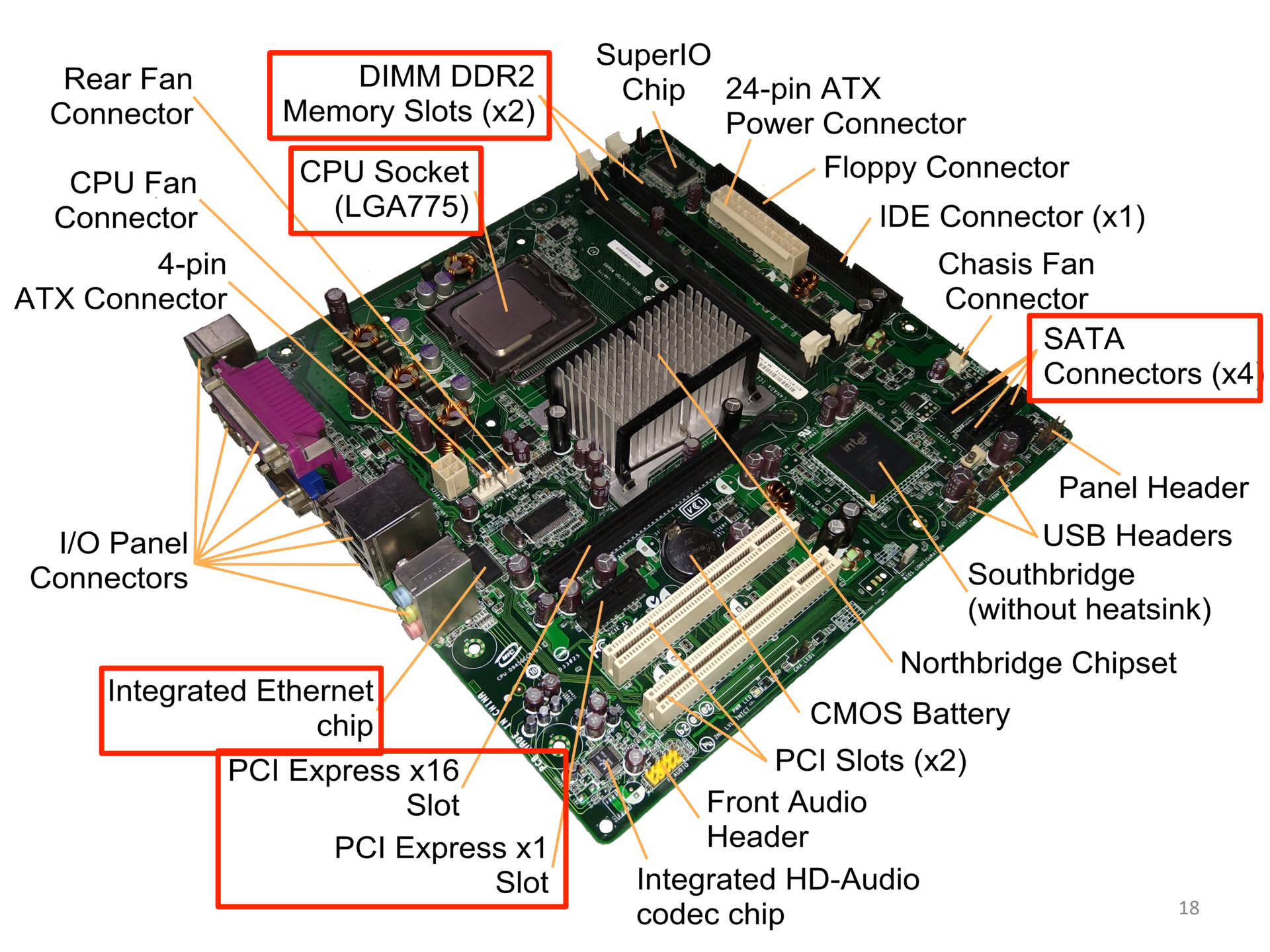
https://www.webopedia.com/TERM/C/computer.html

Key Parts of Computer Hardware

- **Processor** (CPU, GPU, etc.)
  - Hardware to orchestrate and execute *instructions* to manipulate *data* as specified by a *program*
- **Main Memory** (aka Dynamic Random Access Memory)
  - Hardware to store *data* and *programs* that allows very fast location/retrieval; byte-level *addressing* scheme
- **Disk** (aka secondary/persistent storage)
  - Similar to memory but *persistent*, *slower*, and higher capacity / cost ratio; various addressing schemes
- **Network** interface controller (NIC)
  - Hardware to send data to / retrieve data over network of interconnected computers/devices

Abstract Computer Parts and Data




Rear Fan Connector
DIMM DDR2 Memory Slots (x2)
SuperIO Chip
24-pin ATX Power Connector
CPU Fan Connector
CPU Socket (LGA775)
Floppy Connector
IDE Connector (x1)
4-pin ATX Connector
Chasis Fan Connector
SATA Connectors (x4)
Panel Header
USB Headers
I/O Panel Connectors
Southbridge (without heatsink)
Northbridge Chipset
Integrated Ethernet chip
CMOS Battery
PCI Slots (x2)
PCI Express x16 Slot
Front Audio Header
PCI Express x1 Slot
Integrated HD-Audio codec chip

Key Aspects of Software

- **Instruction**
  - A command understood by hardware; finite vocabulary for a processor: Instruction Set Architecture (ISA); bridge between hardware and software
- **Program** (aka code)
  - A collection of instructions for hardware to execute
- **Programming Language (PL)**
  - A human-readable *formal* language to write programs; at a much higher level of *abstraction* than ISA
- **Application Programming Interface (API)**
  - A set of functions ("interface") exposed by a program/set of programs for use by humans/other programs
- **Data**
  - Digital representation of *information* that is stored, processed, displayed, retrieved, or sent by a program

Main Kinds of Software

- **Firmware**
  - Read-only programs “baked into” a device to offer basic hardware control functionalities
- **Operating System (OS)**
  - Collection of interrelated programs that work as an intermediary platform/service to enable application software to use hardware more effectively/easily
  - Examples: Linux, Windows, MacOS, etc.
- **Application Software**
  - A program or a collection of interrelated programs to manipulate data, typically designed for human use
  - Examples: Excel, Chrome, PostgreSQL, etc.

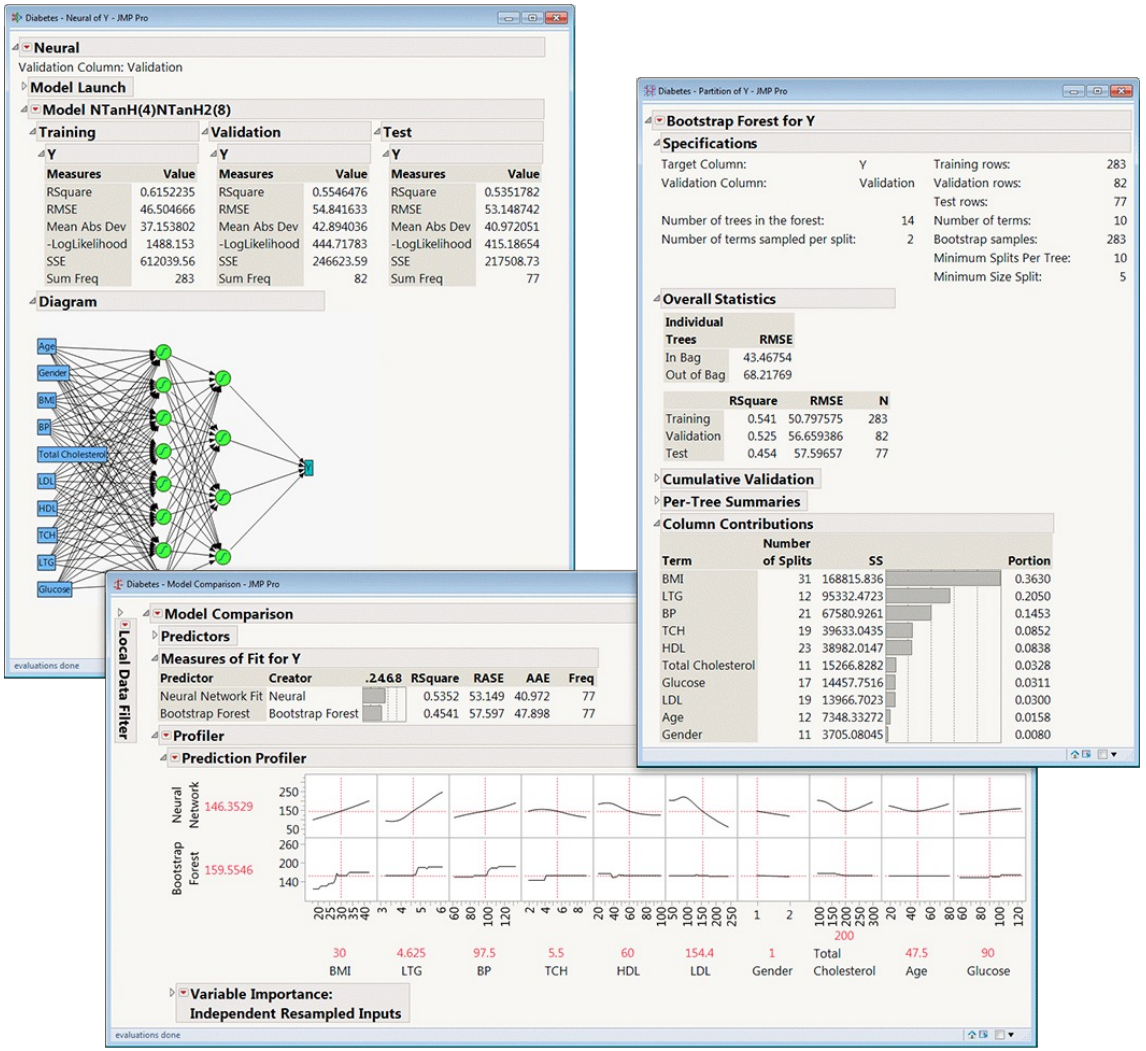
Outline

- Basics of Computer Organization
  - Digital Representation of Data
  - Processors and Memory Hierarchy
- Basics of Operating Systems
  - Process Management: Virtualization; Concurrency
  - Filesystem and Data Files
  - Main Memory Management
- Persistent Data Storage

Q: *But why bother learning such low-level computer sciencey stuff in Data Science?*

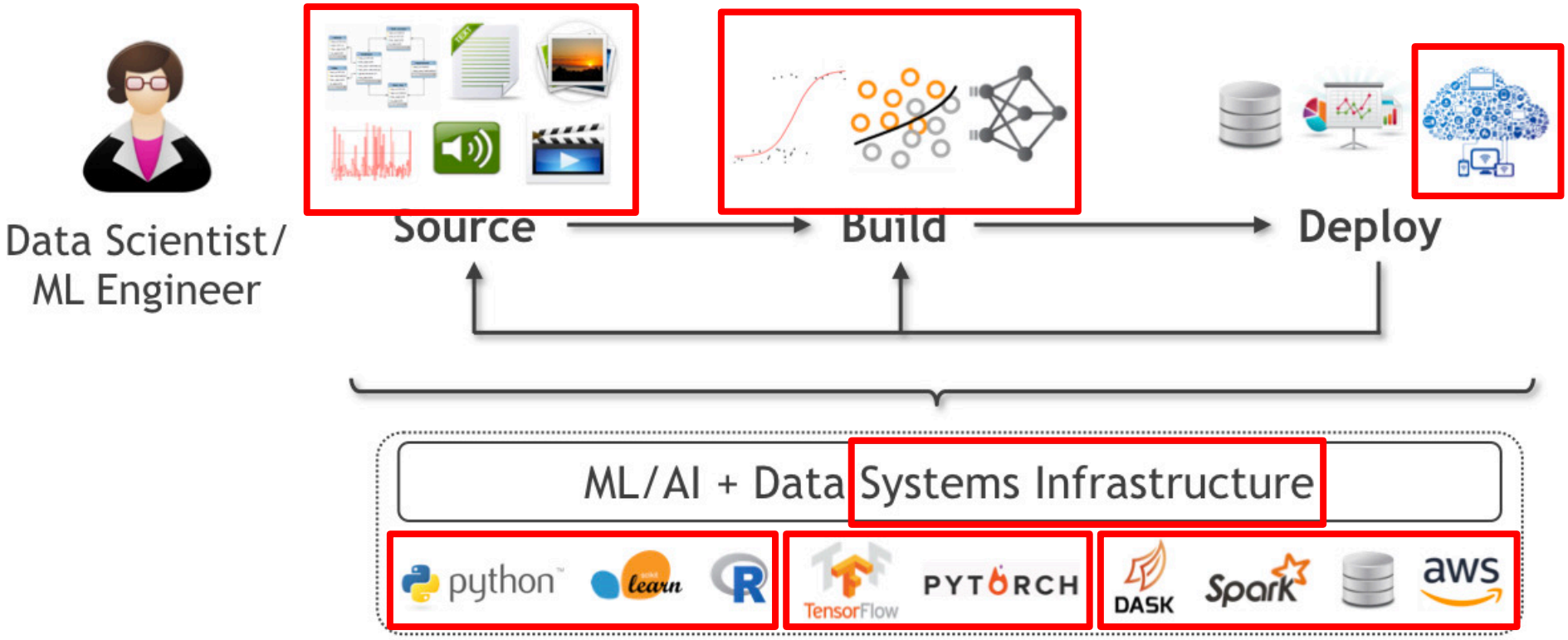
Luxury of “Statisticians”/“Analysts” of Yore

- **Methods:** Sufficed to learn just math/stats, maybe some SQL
- **Types:** Mostly tabular (relational), maybe some time series
- **Scale:** Mostly small (KBs to few GBs)
- **Tools:** Simple GUIs for both analysis and deployment; maybe an R-like console



https://www.jmp.com/en_au/offers/jmp-pro-for-academic-research.html
https://www.technologymagazine.com/data-and-data-analytics/sas-tops-worldwide-advanced-and-predictive-analytics-market-share

Reality of Today's "Data Scientists"



Data acquisition
Data preparation

Feature Engineering
Training & Inference
Model Selection

Serving
Monitoring

Why bother with these in Data Science?

- Basics of Computer Organization
  - Digital Representation of Data
  - Processors and Memory Hierarchy
- Basics of Operating Systems
  - Process Management: Virtualization; Concurrency
  - Filesystem and Data Files
  - Main Memory Management
- Persistent Data Storage

You will face myriad and new data types

Compute hardware is evolving fast

You will need to use new methods on evolving data file formats on clusters / cloud

Storage hardware is evolving fast