FOUNTAINHEAD

“Baruch MFE Big Data in Finance”

~ Class 5 ~

Baruch College

Wednesday 2nd March 2016

FOUNTAINHEAD

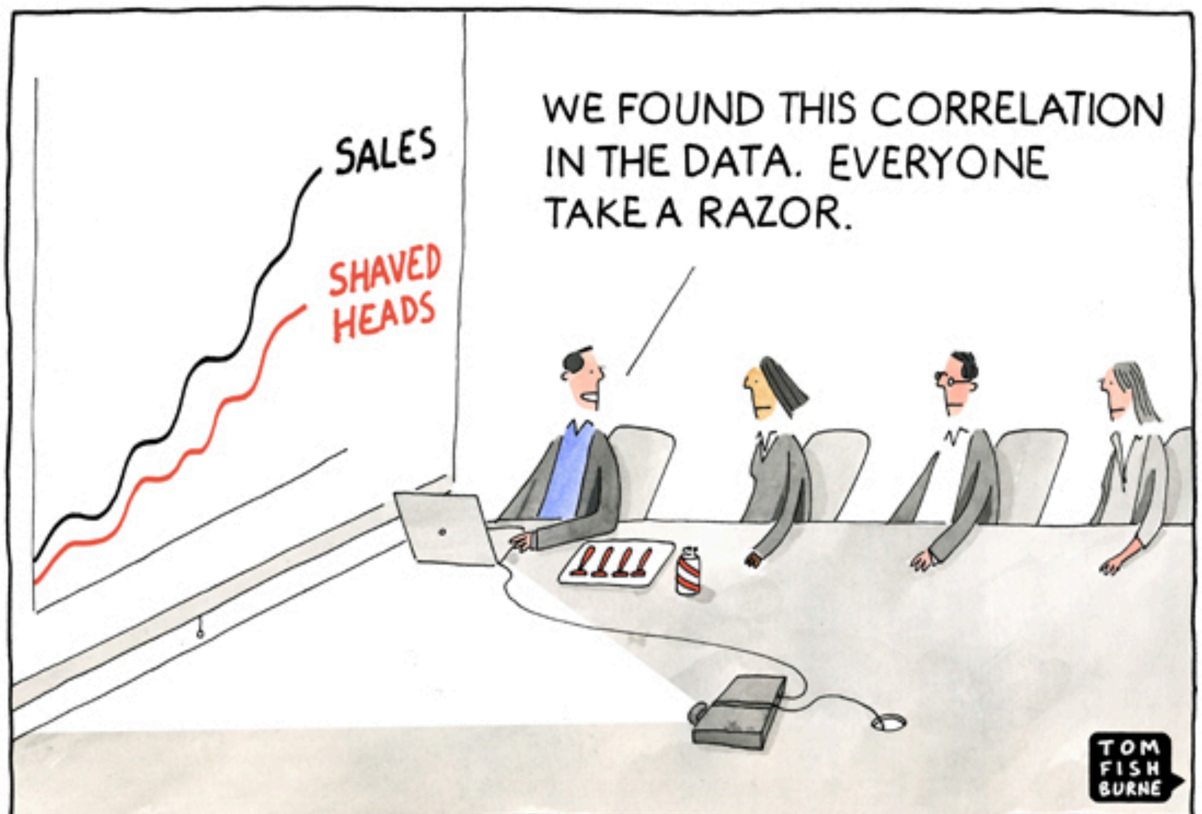
Class 5

In this class we will cover these topics:

1. Review ~ D1 .. D4
2. Basic knowledge ~ Student questions and issues
3. Basic programming knowledge ~ By the numbers
4. Exam preparation: Model answers for 2015 final exam
5. Streaming data. Intro to Apache Spark.
6. Excel as a front-end for Big Data solutions
7. Excel RTD
8. Excel Asynchronous UDF
9. Excel demo and code

FOUNTAINHEAD

Apophenia



© marketoonist.com

FOUNTAINHEAD

D1: Data exploration

Data exploration is made up from these concepts and tools:

1. Classification (5 V's, etc.) [Concept]
2. Math model of your Big Data problem [Concept]
3. Statistical analysis [Concept]
4. Visualization [Concept]
5. Synthetic data [Concept]
6. Custom code (C++, Python, etc.) [Tool]
7. Excel [Tool]
8. R, Matlab, Julia, Mathematica, etc. [Tools]

FOUNTAINHEAD

D2: Data programming

Data programming ("plumbing") is made up from these concepts and tools:

1. Data model (data partitioning, data structures, etc.) [Concept]
2. Execution model (extracting parallelism) [Concept]
3. Parallel programming (designing for scale) [Concept]
4. Data scrubbing (data is a living, breathing thing) [Concept]
5. Instrumentation (log files, notifications, etc.) [Concept]
6. Custom code (C++, Python, etc.) [Tool]
7. Hadoop and other NoSQL databases [Tool]

FOUNTAINHEAD

D3: Data analysis

Data analysis made up from these concepts and tools:

1. Algorithms, Graphs, MapReduce [Concept]
2. Statistical analysis [Concept]
3. Time-series analysis [Concept]
4. Machine learning and data mining [Concept]
5. Optimization [Concept]
6. Custom code (C++, Python, etc.) [Tool]
7. Hadoop and other NoSQL, SAS, etc. [Tool]

FOUNTAINHEAD

D4: Data insights

Data insights is made up from these concepts and tools:

1. Profit from your data. Veracity. Apophenia. [Concept]
2. Machines making decisions [Concept]
3. People making decisions [Concept]
4. Data reporting and visualization. Real-time imperative. [Concept]
5. Custom code (C++, Python, etc.) [Tool]
6. Excel as a front-end [Tool]
7. R, Matlab, Julia, Mathematica, etc. as a front-end [Tools]
8. Mobile as a front-end [Tool]

Basic knowledge

Some thoughts:

1. What does it mean to be professional?
    1. Competent & trustworthy. Example: W. Edwards Demming.
2. What does it mean to be expert?
    1. Basic knowledge.
    2. Knows and uses the tools.
    3. Problem solving skills. Expert knowledge and judgment.
3. What would you like to see if your life was on the line?
    1. Oh, let me just go look that up in a book!

FOUNTAINHEAD

Basic knowledge

Here are some student issues I have encountered:

1. Processes and threads
2. Multi-process (MPI, Hadoop) and multi-thread (C++ threads)
3. Distributed parallelism and program parallelism
4. Techniques: memory mapped files, systems programming
5. Techniques: data storage (structs), data movement (compress)
6. Techniques: synthetic data, structured storage (HDF5)
7. Toolchain (general): Github (vcs), continuous build, documentation
8. Tools (specific): memcheck, profiler, debugger, test scripts

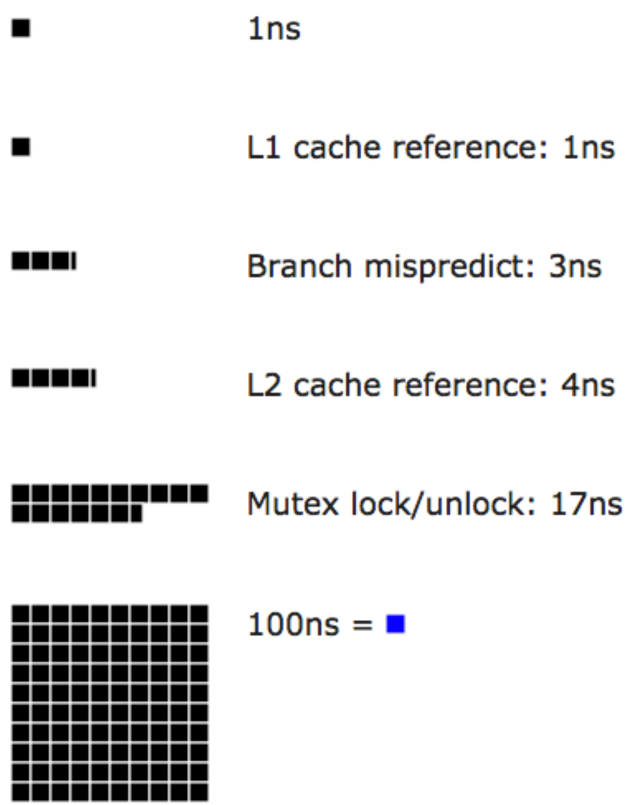
FOUNTAINHEAD

Basic programming knowledge

```
Latency Comparison Numbers
--------------------------
L1 cache reference                           0.5 ns
Branch mispredict                            5   ns
L2 cache reference                           7   ns                      14x L1 cache
Mutex lock/unlock                           25   ns
Main memory reference                      100   ns                      20x L2 cache, 200x L1 cache
Compress 1K bytes with Zippy             3,000   ns
Send 1K bytes over 1 Gbps network       10,000   ns    0.01 ms
Read 4K randomly from SSD*             150,000   ns    0.15 ms
Read 1 MB sequentially from memory     250,000   ns    0.25 ms
Round trip within same datacenter      500,000   ns    0.5  ms
Read 1 MB sequentially from SSD*     1,000,000   ns    1    ms  4X memory
Disk seek                           10,000,000   ns   10    ms  20x datacenter roundtrip
Read 1 MB sequentially from disk    20,000,000   ns   20    ms  80x memory, 20X SSD
Send packet CA->Netherlands->CA    150,000,000   ns  150    ms
```

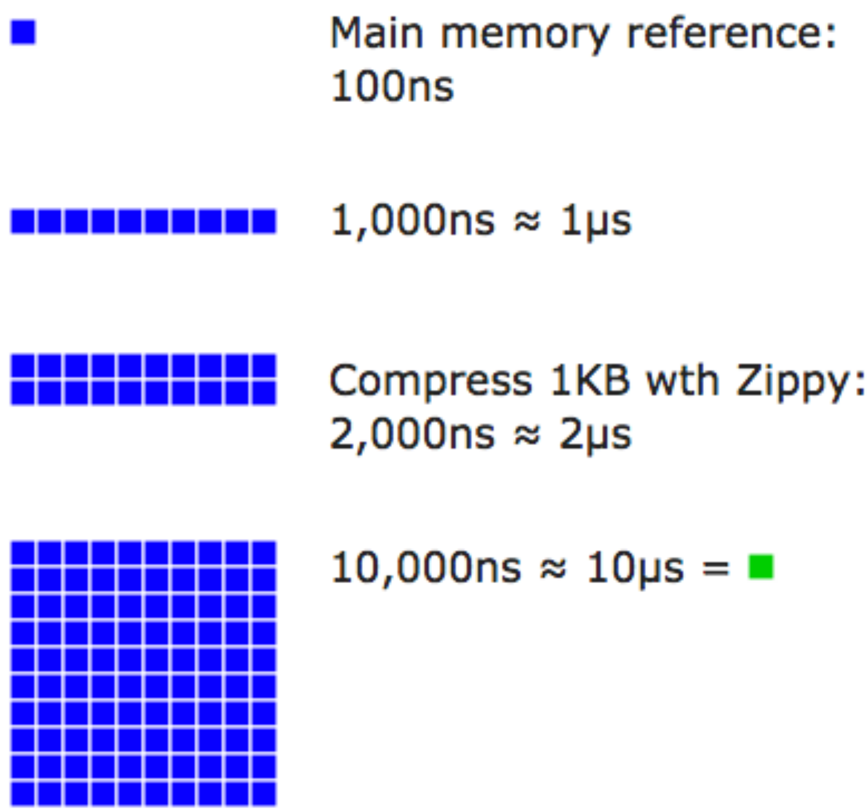
FOUNTAINHEAD

Basic programming knowledge



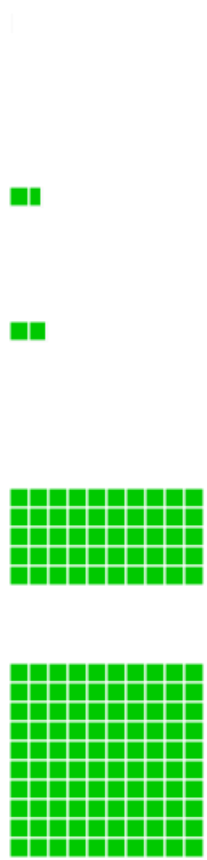
Fountainhead

Basic programming knowledge



FOUNTAINHEAD

Basic programming knowledge



Send 2,000 bytes over commodity network: 1,000ns ≈ 0.7μs

SSD random read: 16,000ns ≈ 16μs

Read 1,000,000 bytes sequentially from memory: 19,000ns ≈ 19μs

Round trip in same datacenter: 500,000ns ≈ 500μs

1,000,000ns = 1ms = ■

FOUNTAINHEAD

Basic programming knowledge

Read 1,000,000 bytes sequentially from SSD: 300,000ns

Disk seek: 4,000,000ns ≈ 4ms

Read 1,000,000 bytes sequentially from disk: 2,000,000ns ≈ 2ms

Packet roundtrip CA to Netherlands: 150,000,000ns ≈ 150ms

FOUNTAINHEAD

Basic programming knowledge

Some more numbers that are very useful to know (as of 2015):

1. CPU: 10's of threads. 2nJ per op. 8 cores. TB RAM. 25GB/s. 31GB/s.
2. GPU: 1,000's of threads. 200pJ per op. 4992 cores. 24GB. 480GB/s.
3. PCIe bus: 5GB/s, but getting better.
4. Hard disk: 4TB. 6-7 Watts. $0.15/GB. 120MB/s I/O.
5. SSD disk: 1TB. 2-3 Watts. $0.50/GB. 550MB/s I/O.

FOUNTAINHEAD

Big Data Streaming

It lives!!!

Big Data Streaming

The reality of Big Data:

1. Nearly all data sets are living, breathing things.
2. You should think of your data stretching far into the past, and further into the future.
3. Continuous flowing or streaming of data.
4. (That's why D2: Data programming is so important!)
5. Solution is to think of your data as a flowing stream.
6. Data streaming.
7. Data pipelines.

Big Data Streaming

Characteristics of Big Data streams:

1. Continuous, ordered in time, but sometimes changing, fast, cumulatively huge in size. Unpredictable arrival times and rates.
2. (Quick quiz: relationship to 5 V's?)
3. Need to think of your data set as infinite. New algos, e.g. MinHash.
4. Data "plumbing" has to be completely automated (D2). Max velocity.
5. Fast changes require fast, real-time response.
6. Random access very expensive. Can only really look as it passes by.
7. Store it all? Or store only a summary of past data. ½ life recent data.

Fountainhead

Big Data Streaming

Examples:

1. Traditional time-series market data. But no longer just EOD. Ticks.
2. Credit card transactions.
3. Bitcoin transactions on the blockchain.
4. Internal data for real-time risk, e.g. CVA.
5. Twitter stream for sentiment and event-driven investing.
6. News feeds (news bots!).
7. Dark pools and many trading venues. More venues, more data.
8. Electronic markets means trading moving to 24x7 trading.

Fountainhead

Big Data Streaming

Practicalities:

1. Need to design and build for maximum data arrival rate, not the average rate. Otherwise, be prepared to throw away data.
2. Getting data into an out of memory. Small things make a difference.
3. Continuous querying of data. Answers updated over time. Windows.
4. Need to communicate with humans via a good front-end. (Machines.)
5. Fixed memory size algorithms, regardless of data size.
6. With bounded memory and partial view of data, answers are approx.
7. Data reduction and synopsis construction methods: e.g. sub-linearity.

FOUNTAINHEAD

Big Data Streaming ~ Spark

Apache Spark for Big Data streaming:

1. Apache open source project.
2. Key components …
3. Spark Core and Resilient Distributed Datasets (RDDs).
4. Spark SQL.
5. Spark Streaming.
6. MLlib Machine Learning Library.
7. GraphX.
8. Java, Scala, and Python APIs.