Unpredictable Code

Branch Prediction
Jump Prediction
Predication
Control code (ifs)

Partitioning into cache size chunks


RAM
Random Access?
Fast?
Volatile
Expensive (more than 100x in 2013)
Physically Addressed

How Fast? (In CPU cycles)
How Random? 64 Byte EDO burst size

Memory Pages
CPUs
Address Translation

How many additions per second?
Many Cores
Caches
Pipelined
Cache Size?
Complicated & Expensive
Cacheable
Designated Address Cache

What is the length of a CPU pipeline? 15 to 20 stages
How many pipeline bubbles per second?
3.125 GHz * 4 Pipelines * 8 Byte = 100 GB/s
1600 GT/s * 4 Channels * 8 Byte ≈ 50 GB/s
CPU twice as fast as RAM
100x faster than disk
100 GB/s by 20 = 5 GB/s

How do Pipeline bubbles occur?
Function calls - Slide
What about Iterators?
False Sharing
How to remove iterators?
Are we good? No!
How is that solved in the paper?
Furthermore, operators are connected by materializing intermediate results as temporary tables inside the buffer pool and streaming them to subsequent operators.

What are the cost for Materialization?
No!
Is this it? Best achievable performance?

no if(input[i].value > 5)
output[j++] = input[i]
output[j] = input[i]
j+=(input[i].value > 5)

Let's say for 100 GB of data passing through four predicated selections + aggregate
Vector-Wise Processing (Fastest in memory TPCH)

How Large Should Vectors be?
Empirical value: half L1

Efficent Hashing
Caching
Recycling
Prefetching
NVRAM

Results

Things we didn't talk about
jitt
EMM
GPHS
Multicore
Resoning
Efficient licensing
Caching Effects
Prefecting
NVIDIA