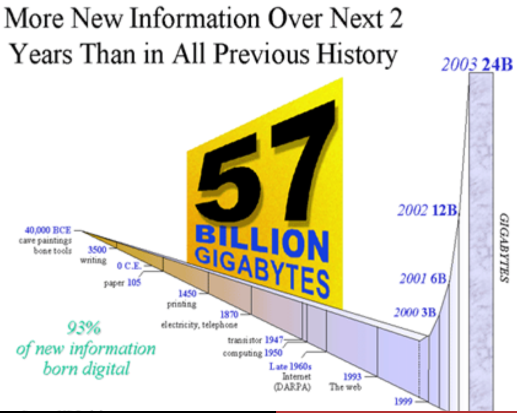
Massive Data Algorithmics

Lecture 1: Introduction

Massive Data

- Massive datasets are being collected everywhere
- Storage management software is billion-dollar industry

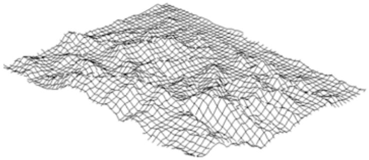


Examples

- Phone: AT&T 20TB phone call database, wireless tracking
- Consumer: WalMart 70TB database, buying patterns
- WEB: Web crawl of 200M pages and 2000M links, Akamai stores 7 billion clicks per day
- Geography: NASA satellites generate 1.2TB per day

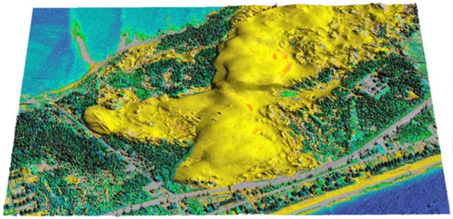
Grid Terrain Data

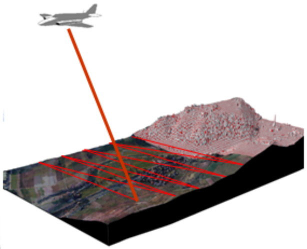
- Appalachian Mountains (800km x 800km)
    - 100m resolution ⇒ ∼64M cells ⇒ ∼128MB raw data (∼500MB when processing)
    - ∼1.2GB at 30m resolution NASA SRTM mission acquired 30m data for 80% of the earth land mass
    - ∼12GB at 10m resolution (much of US available from USGS)
    - ∼1.2TB at 1m resolution (selected, mostly military)



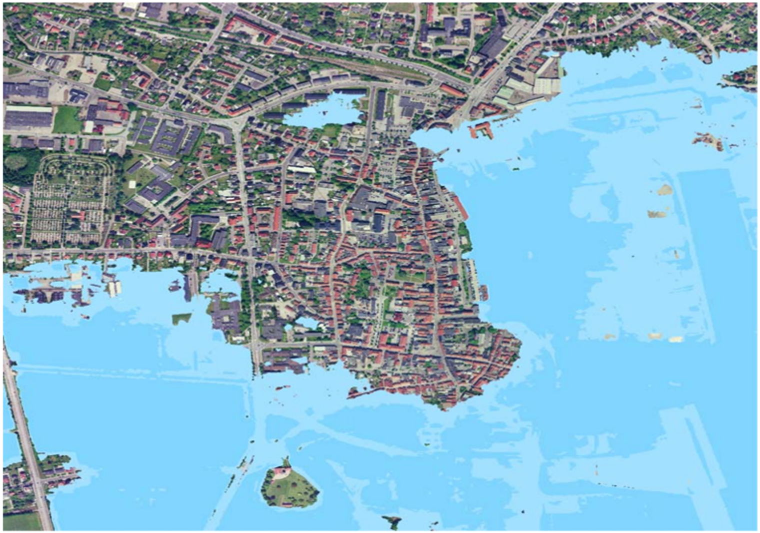
LIDAR Terrain Data

- Massive (irregular) point sets (1-10m resolution)
- Appalachian Mountains between 50GB and 5TB

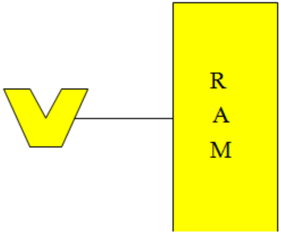




Application Example: Flooding Prediction

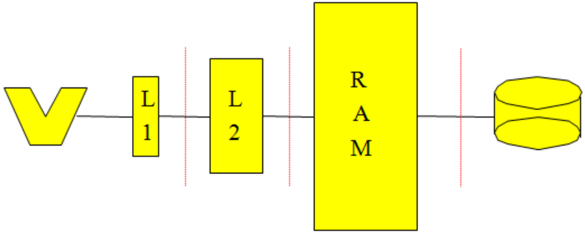


Random Access Machine Model



- Standard theoretical model of computation:
  - Infinite memory
  - Uniform access cost
- Simple model crucial for success of computer industry

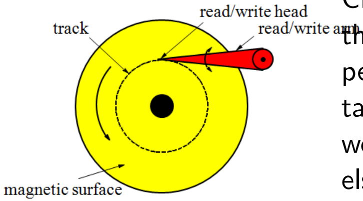
Hierarchical Memory



- Modern machines have complicated memory hierarchy
    - Levels get larger and slower further away from CPU
    - Data moved between levels using large blocks

Slow IO

- Disk access is 10^6 times slower than main memory access

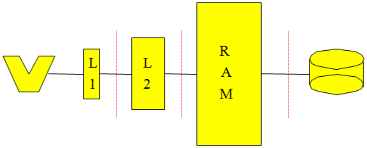


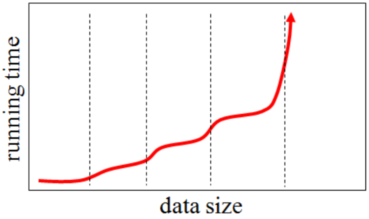
The difference in speed between modern CPU and disk technologies is analogous to the difference in speed in sharpening a pencil using a sharpener on ones desk or by taking an airplane to the other side of the world and using a sharpener on someone elses desk. (D. Comer)

- Disk systems try to amortize large access time transferring large contiguous blocks of data (8-16Kbytes)
- Important to store/access data to take advantage of blocks (locality)

Scalability Problems

- Most programs developed in RAM-model. Run on large datasets because OS moves blocks as needed
- Moderns OS utilizes sophisticated paging and prefetching strategies. But if program makes scattered accesses even good OS cannot take advantage of block access





External Memory Model(Cache-Aware Model)

$N =$ # of items in the problem instance

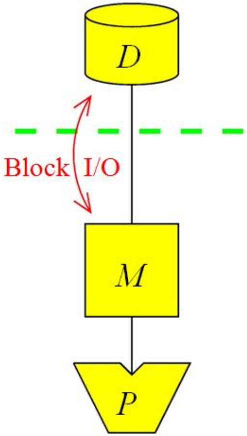
$B =$ # of items per disk block

$M =$ # of items that fit in main memory

$T =$ # of items in output

I/O: Move block between memory and disk

We assume (for convenience) that $M > B^2$



Fundamental Bounds

	Internal	External
Scanning	N	N/B
Sorting	$N \log N$	$N/B \log_{M/B} N/B$
Permuting	N	$\min(N, N/B \log_{M/B} N/B)$
Searching	$\log N$	$\log_B N$

Note:

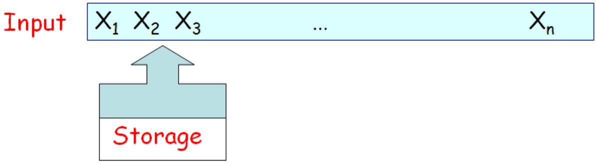
- Linear I/O: $O(N/B)$
- Permuting not linear
- Permuting and sorting bounds are equal in all practical cases
- B factor VERY important: $N/B < (N/B)\log_{M/B}(N/B) << N$

Cache-Oblivious Model

- A cache-oblivious algorithm is an algorithm designed to take advantage of a CPU cache without having the size of the cache
- a cache oblivious algorithm is designed to perform well, without modification, on multiple machines with different cache sizes, or for a memory hierarchy with different levels of cache having different sizes.
- The idea for cache-oblivious algorithms was conceived by Charles E. Leiserson as early as 1996 and first published by Harald Prokop in his master's thesis at the Massachusetts Institute of Technology in 1999.

Streaming Model

In stream model, input data are not available for random access from disk or memory, but rather arrive as one or more continuous data streams.



Performance of algorithm is measured by three basic factors:

- Number of passes algorithm must make over stream.
- The available memory.
- The running time of the algorithm.

Grading

- Midterm+ Final: 15 points
- Homework, Project, Presentation: 5 points