Introduction to Streaming Algorithms

Jeff M. Phillips

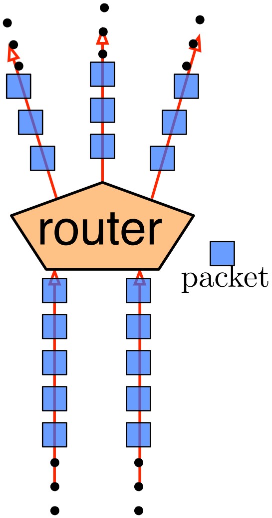
September 21, 2013

Network Router

Internet Router

- data per day: at least I Terabyte
- packet takes 8 nanoseconds to pass through router
- few million packets per second

What statistics can we keep on data?
Want to detect anomalies for security.

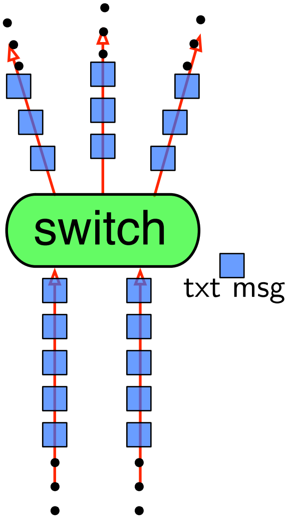


Telephone Switch

Cell phones connect through switches

- each message 1000 Bytes
- 500 Million calls / day
- 1 Terabyte per month

Search for characteristics for dropped calls?

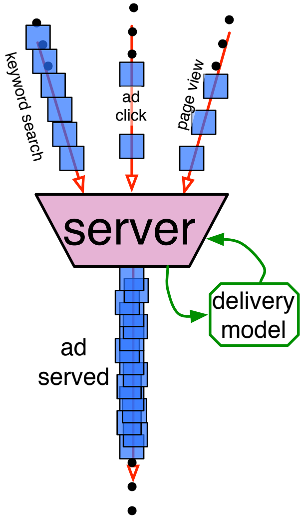


Ad Auction

Serving Ads on web
Google, Yahoo!, Microsoft

- Yahoo.com viewed 100 trillion times
- 2 million / hour
- Each page serves ads; which ones?

How to update ad delivery model?

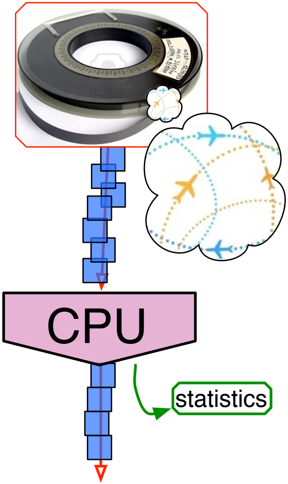


Flight Logs on Tape

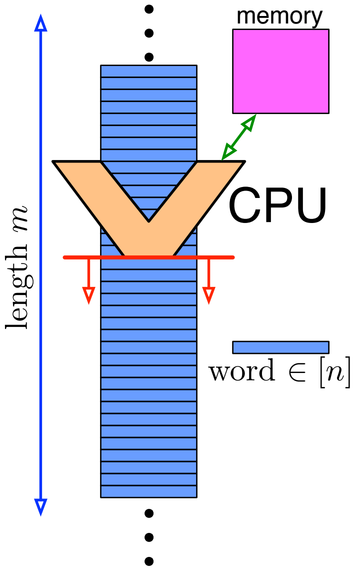
All airplane logs over Washington, DC

- About 500 - 1000 flights per day.
- 50 years, total about 9 million flights
- Each flight has trajectory, passenger count, control dialog

Stored on Tape. Can make 1 pass!
What statistics can be gathered?



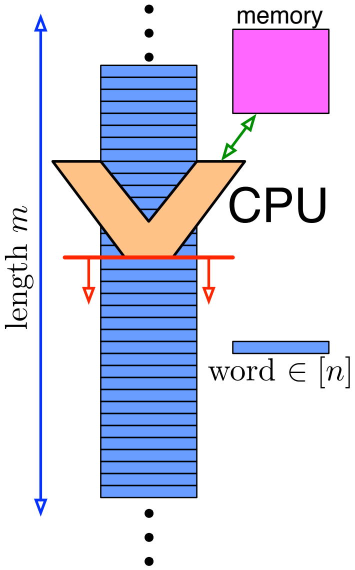
Streaming Model



CPU makes "one pass" on data

- Ordered set $A = \langle a_1, a_2, \ldots, a_m \rangle$
- Each $a_i \in [n]$, size $\log n$
- Compute $f(A)$ or maintain $f(A_i)$ for $A_i = \langle a_1, a_2, \ldots, a_i \rangle$.

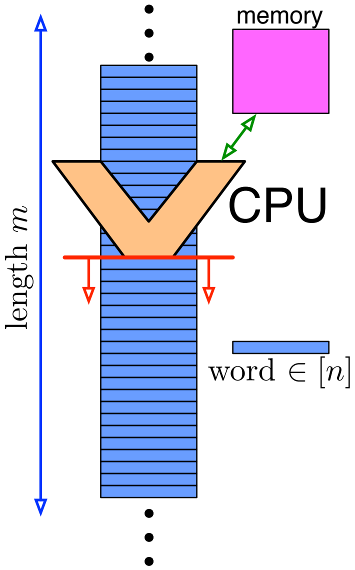
Streaming Model



CPU makes "one pass" on data

- Ordered set $A = \langle a_1, a_2, \ldots, a_m \rangle$
- Each $a_i \in [n]$, size $\log n$
- Compute $f(A)$ or maintain $f(A_i)$ for $A_i = \langle a_1, a_2, \ldots, a_i \rangle$.
- Space restricted to $S = O(\text{poly}(\log m, \log n))$.
- Updates $O(\text{poly}(S))$ for each a_i.

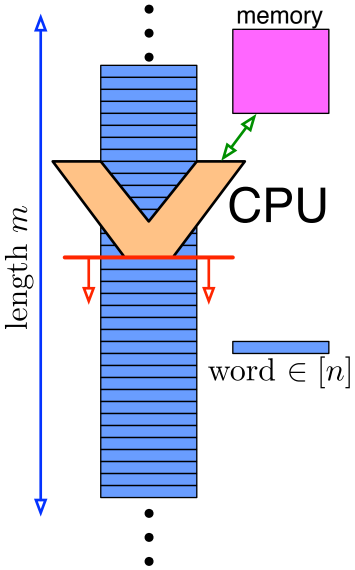
Streaming Model



Space:

- Ideally $S = O(\log m + \log n)$
- $\log n =$ size of 1 word
- $\log m =$ to store number of words

Streaming Model



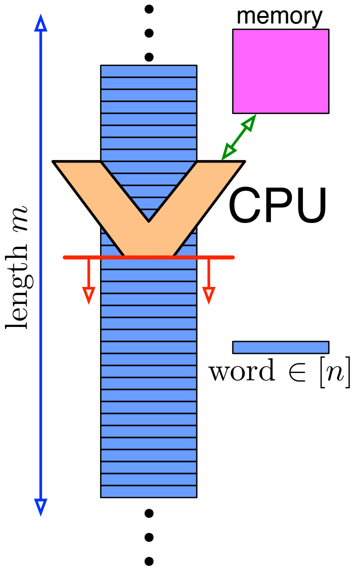
Space:

- Ideally $S = O(\log m + \log n)$
- $\log n$ = size of 1 word
- $\log m$ = to store number of words

Updates:

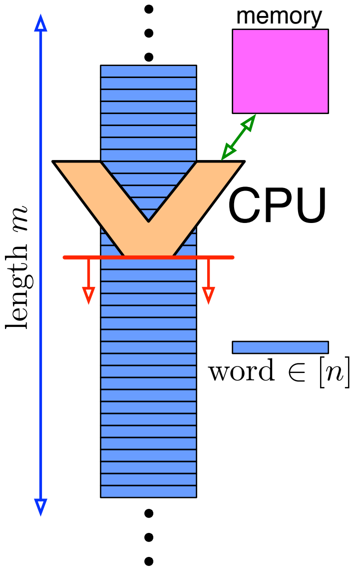
- $O(S^2)$ or $O(S^3)$ can be too much!
- Ideally updates in $O(S)$

Easy Example: Average



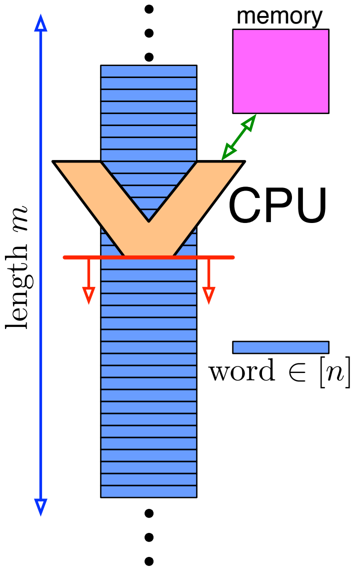
- Each a_i a number in $[n]$
- $f(A_i) = \text{average}(\{a_1, \ldots, a_i\})$

Easy Example: Average



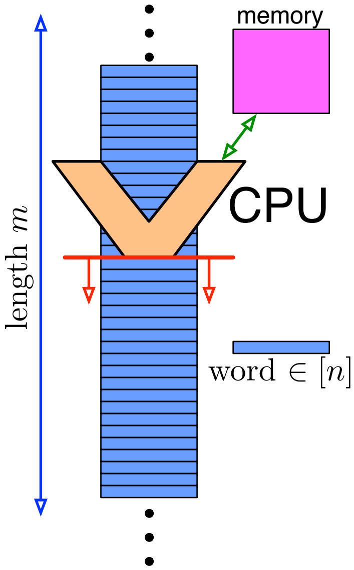
- Each a_i a number in $[n]$
- $f(A_i) = \text{average}(\{a_1, \ldots, a_i\})$
- Maintain: i and $s = \sum_{j=1}^{i} a_i$.
- $f(A_i) = s/i$

Easy Example: Average



- Each a_i a number in $[n]$
- $f(A_i) = \text{average}(\{a_1, \ldots, a_i\})$
- Maintain: i and $s = \sum_{j=1}^{i} a_i$.
- $f(A_i) = s/i$
- Problem? s is bigger than a word!

Easy Example: Average



- Each a_i a number in $[n]$
- $f(A_i) = \text{average}(\{a_1, \ldots, a_i\})$
- Maintain: i and $s = \sum_{j=1}^{i} a_i$.
- $f(A_i) = s/i$
- Problem? s is bigger than a word!
- s is not bigger than $(\log s / \log n)$ words (big int data structure)
- usually 2 or 3 words is fine

Trick 1: Approximation

Return $\hat{f}(A)$ instead of $f(A)$ where

$$|f(A) - \hat{f}(A)| \leq \varepsilon \cdot f(A).$$

$\hat{f}(A)$ is a $(1+\varepsilon)$-*approximation* of $f(A)$.

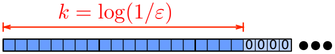
Trick 1: Approximation

Return $\hat{f}(A)$ instead of $f(A)$ where

$$|f(A) - \hat{f}(A)| \leq \varepsilon \cdot f(A).$$

$\hat{f}(A)$ is a $(1+\varepsilon)$-*approximation* of $f(A)$.

Example: Average



- (a) the count: i
- (b) top $k = \log(1/\varepsilon) + 1$ bits of s: $\hat{s}$
- (c) number of bits in s
- Let $\hat{f}(A) = \hat{s}/i$

Trick 1: Approximation

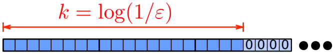
Return $\hat{f}(A)$ instead of $f(A)$ where

$$|f(A) - \hat{f}(A)| \leq \varepsilon \cdot f(A).$$

$\hat{f}(A)$ is a $(1+\varepsilon)$-*approximation* of $f(A)$.

Example: Average

- (a) the count: i
- (b) top $k = \log(1/\varepsilon) + 1$ bits of s: $\hat{s}$
- (c) number of bits in s
- Let $\hat{f}(A) = \hat{s}/i$



First bit has $\geq (1/2)f(A)$
Second bit has $\leq (1/4)f(A)$
jth bit has $\leq (1/2^j)f(A)$

$$\sum_{j=k+1}^{\infty} (1/2^j)f(A) < (1/2^k)f(A) < \varepsilon \cdot f(A)$$

Trick 1: Approximation

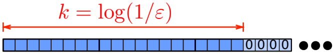
Return $\hat{f}(A)$ instead of $f(A)$ where

$$|f(A) - \hat{f}(A)| \leq \varepsilon \cdot f(A).$$

$\hat{f}(A)$ is a $(1+\varepsilon)$-*approximation* of $f(A)$.

Example: Average

- (a) the count: i
- (b) top $k = \log(1/\varepsilon) + 1$ bits of s: $\hat{s}$
- (c) number of bits in s
- Let $\hat{f}(A) = \hat{s}/i$



First bit has $\geq (1/2)f(A)$
Second bit has $\leq (1/4)f(A)$
jth bit has $\leq (1/2^j)f(A)$

$$\sum_{j=k+1}^{\infty} (1/2^j)f(A) < (1/2^k)f(A) < \varepsilon \cdot f(A)$$

Where did I cheat?

Trick 2: Randomization

Return $\hat{f}(A)$ instead of $f(A)$ where

$$\mathbf{Pr}\left[|f(A) - \hat{f}(A)| > \varepsilon \cdot f(A)\right] \leq \delta.$$

$\hat{f}(A)$ is a $(1 + \varepsilon, \delta)$-*approximation* of $f(A)$.

Trick 2: Randomization

Return $\hat{f}(A)$ instead of $f(A)$ where

$$\mathbf{Pr}\left[|f(A) - \hat{f}(A)| > \varepsilon \cdot f(A)\right] \leq \delta.$$

$\hat{f}(A)$ is a $(1+\varepsilon, \delta)$-*approximation* of $f(A)$.

Can fix previous cheat using randomization and Morris Counter (Morris 78, Flajolet 85)

Decreasing Probability of Failure

Investment Company (IC) sends out 100×2^k emails:

- 2^{k-1} say Stock A will go up in next week
- 2^{k-1} say Stock A will go down in next week

After 1 week, 1/2 of email receivers got good advice.

Decreasing Probability of Failure

Investment Company (IC) sends out 100×2^k emails:

- 2^{k-1} say Stock A will go up in next week
- 2^{k-1} say Stock A will go down in next week

After 1 week, 1/2 of email receivers got good advice.

Next week, IC sends letters 2^{k-1} letters, only to those who got good advice.

- 2^{k-2} say Stock B will go up in next week.
- 2^{k-2} say Stock B will go down in next week.

After 2 weeks, 1/4 of all receivers have gotten good advice twice.

Decreasing Probability of Failure

Investment Company (IC) sends out 100×2^k emails:

- 2^{k-1} say Stock A will go up in next week
- 2^{k-1} say Stock A will go down in next week

After 1 week, $1/2$ of email receivers got good advice.

Next week, IC sends letters 2^{k-1} letters, only to those who got good advice.

- 2^{k-2} say Stock B will go up in next week.
- 2^{k-2} say Stock B will go down in next week.

After 2 weeks, $1/4$ of all receivers have gotten good advice twice.

After k weeks 100 receivers got good advice

- IC now asks each for money to receive future stock tricks.

Decreasing Probability of Failure

Investment Company (IC) sends out 100×2^k emails:

- 2^{k-1} say Stock A will go up in next week
- 2^{k-1} say Stock A will go down in next week

After 1 week, $1/2$ of email receivers got good advice.

Next week, IC sends letters 2^{k-1} letters, only to those who got good advice.

- 2^{k-2} say Stock B will go up in next week.
- 2^{k-2} say Stock B will go down in next week.

After 2 weeks, $1/4$ of all receivers have gotten good advice twice.

After k weeks 100 receivers got good advice

- IC now asks each for money to receive future stock tricks.
- Don't actually do this!!!

Decreasing Probability of Failure

Investment Company (IC) sends out 100×2^k emails:

- 2^{k-1} say Stock A will go up in next week
- 2^{k-1} say Stock A will go down in next week

After 1 week, $1/2$ of email receivers got good advice.

Next week, IC sends letters 2^{k-1} letters, only to those who got good advice.

- 2^{k-2} say Stock B will go up in next week.
- 2^{k-2} say Stock B will go down in next week.

After 2 weeks, $1/4$ of all receivers have gotten good advice twice.

After k weeks 100 receivers got good advice

- IC now asks each for money to receive future stock tricks.
- Don't actually do this!!!

If you are on IC's original email list, with what probability will you *not* receive k good stock tips?

Decreasing Probability of Failure

Investment Company (IC) sends out 100×2^k emails:

- 2^{k-1} say Stock A will go up in next week
- 2^{k-1} say Stock A will go down in next week

After 1 week, 1/2 of email receivers got good advice.

Next week, IC sends letters 2^{k-1} letters, only to those who got good advice.

- 2^{k-2} say Stock B will go up in next week.
- 2^{k-2} say Stock B will go down in next week.

After 2 weeks, 1/4 of all receivers have gotten good advice twice.

After k weeks 100 receivers got good advice

- IC now asks each for money to receive future stock tricks.
- Don't actually do this!!!

If you are on IC's original email list, with what probability will you *not* receive k good stock tips?

$$1 - (1/2)^k$$

Sliding Window

Let $A_{i-s,i} = \{a_{i-s}, a_{i-s+1}, \ldots, a_i\}$ (the last s items).
Goal: maintain $f(A_{i-s,i})$.

Sliding Window

Let $A_{i-s,i} = \{a_{i-s}, a_{i-s+1}, \ldots, a_i\}$ (the last s items).
Goal: maintain $f(A_{i-s,i})$.

Another model: Each $a_i = (v, t)$ where t is a time stamp.
Let $A_i^{[w]} = \{a = (v, t) \in A_i \mid t \geq t_{\text{now}} - w\}$
Goal: maintain $f(A_i^{[w]})$.

Sliding Window

Let $A_{i-s,i} = \{a_{i-s}, a_{i-s+1}, \ldots, a_i\}$ (the last s items).
Goal: maintain $f(A_{i-s,i})$.

Another model: Each $a_i = (v, t)$ where t is a time stamp.
Let $A_i^{[w]} = \{a = (v, t) \in A_i \mid t \geq t_{\text{now}} - w\}$
Goal: maintain $f(A_i^{[w]})$.

Simpler solution: Decay rate γ.
Maintain a summary $S_i = f(A_i)$;
at each time step update $S_{i+1} = f((1-\gamma)S_i \cup a_{i+1})$.

Semi-Streaming Model

Streaming on Graphs.

Semi-Streaming Model

Streaming on Graphs.

Each $a_i = (v_i, v_j)$ is an edge.

- Is graph connected?
- Size of best matching? (each vertex in at most one pair)

Semi-Streaming Model

Streaming on Graphs.

Each $a_i = (v_i, v_j)$ is an edge.

- Is graph connected?
- Size of best matching? (each vertex in at most one pair)

Too hard!

Assume that all vertices can fit in memory, say $O(n \log n)$ space.
For 1 million vertices, may be ok,
but not for 1 billion vertices (e.g. Facebook).

Markov Inequality

Let X be a random variable (RV).
Let $a > 0$ be a parameter.

$$\mathbf{Pr}\left[|X| \geq a\right] \leq \frac{\mathbf{E}[|X|]}{a}.$$

Chebyshev's Inequality

Let Y be a random variable.
Let $b > 0$ be a parameter.

$$\mathbf{Pr}\left[|Y - \mathbf{E}[Y]| \geq b\right] \leq \frac{\mathbf{Var}[|Y|]}{b^2}.$$

Chernoff Inequality

Let $\{X_1, X_2, \ldots, X_r\}$ be independent random variables.
Let $\Delta_i = \max\{X_i\} - \min\{X_i\}$.
Let $M = \sum_{i=1}^{r} X_i$.
Let $\alpha > 0$ be a parameter.

$$\mathbf{Pr}\left[|M - \sum_{i=1}^{r} \mathbf{E}[X_i]| \geq \alpha \right] \leq 2 \exp\left(\frac{-2\alpha^2}{\sum_i \Delta_i^2} \right)$$

Chernoff Inequality

Let $\{X_1, X_2, \ldots, X_r\}$ be independent random variables.
Let $\Delta_i = \max\{X_i\} - \min\{X_i\}$.
Let $M = \sum_{i=1}^{r} X_i$.
Let $\alpha > 0$ be a parameter.

$$\mathbf{Pr}\left[|M - \sum_{i=1}^{r} \mathbf{E}[X_i]| \geq \alpha \right] \leq 2\exp\left(\frac{-2\alpha^2}{\sum_i \Delta_i^2} \right)$$

Often: $\Delta = \max_i \Delta_i$ and $\mathbf{E}[X_i] = 0$ then:

$$\mathbf{Pr}\left[|M| \geq \alpha \right] \leq 2\exp\left(\frac{-2\alpha^2}{r\Delta_i^2} \right)$$

Attribution

These slides borrow from material by Muthu Muthukrishnan:
`http://www.cs.mcgill.ca/~denis/notes09.pdf`
and Amit Chakrabarti:
`http://www.cs.dartmouth.edu/~ac/Teach/CS85-Fall09/`