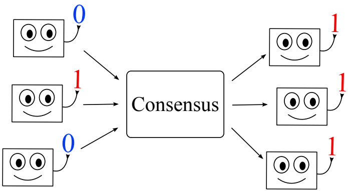
Randomized consensus in expected $O(n^2)$ total work using single-writer registers

James Aspnes
Yale

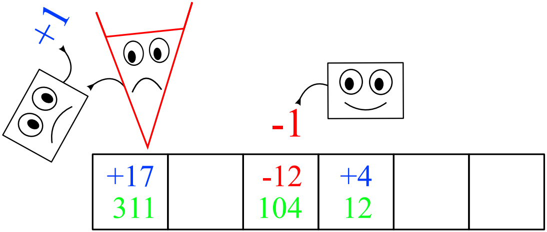
September 20th, 2011

Consensus



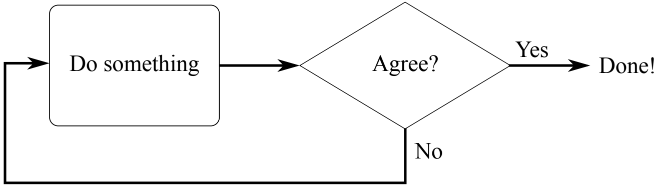
- **Termination:** All non-faulty processes terminate.
- **Validity:** Every output value is somebody's input.
- **Agreement:** All output values are equal.

Asynchronous single-writer register model



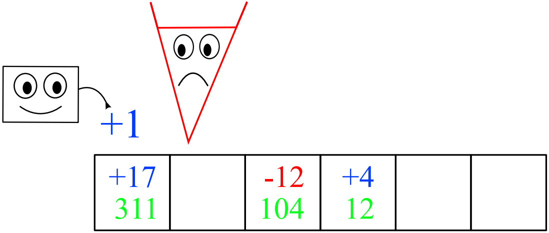
- *n* concurrent processes.
- Each can write to its own register.
- Timing controlled by an **adversary scheduler**.
- Algorithm is **wait-free**: tolerates $n - 1$ **crash failures**.

Implementing consensus



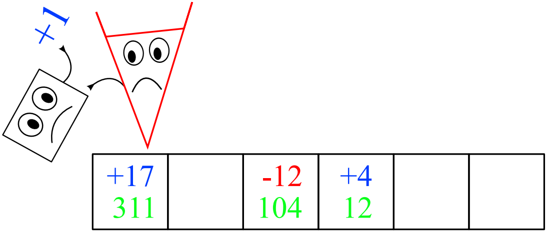
- Typical implementation: use some randomized process that produces agreement with some probability, and commit to a return value when we detect agreement.
- **Weak shared coin** chooses each value $\{0, 1\}$ with probability at least δ.
- If δ is constant, expected cost of consensus $=$ O(cost of weak shared coin). (Aspnes and Herlihy, 1990)

How to build a weak shared coin



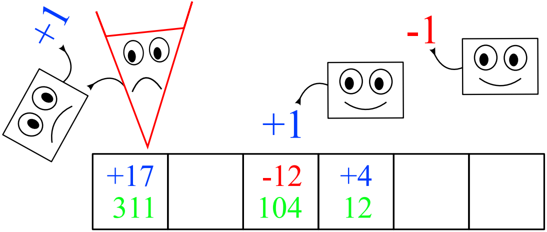
- Take majority of many ± 1 random votes.
- Adversary can stop up to $n - 1$ of them.
- But we generate $\Theta(n^2)$ votes.
- So majority is not affected (with constant probability).

How to build a weak shared coin



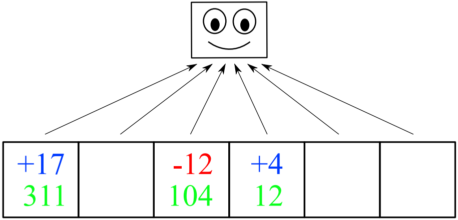
- Take majority of many ± 1 random votes.
- Adversary can stop up to $n-1$ of them.
- But we generate $\Theta(n^2)$ votes.
- So majority is not affected (with constant probability).

How to build a weak shared coin



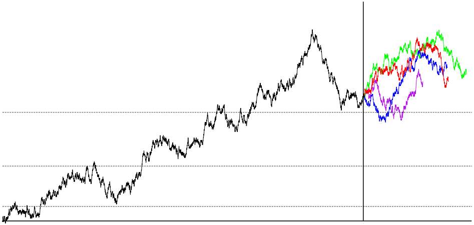
- Take majority of many ± 1 random votes.
- Adversary can stop up to $n - 1$ of them.
- But we generate $\Theta(n^2)$ votes.
- So majority is not affected (with constant probability).

Collecting the votes



- Total vote is computed by reading all registers (a **collect**).
- Collects are expensive ($\Theta(n)$ operations), so we can't do them very often.

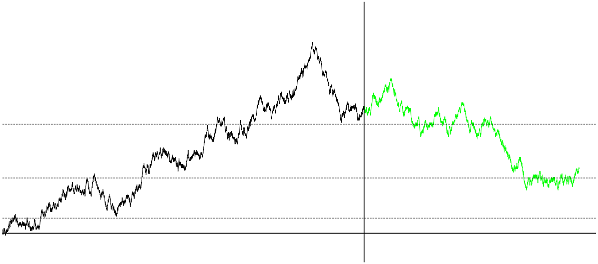
Bracha-Rachman protocol



- Check total every $\Theta(n/\log n)$ votes.
- $\Rightarrow$ Amortized work per vote is $\Theta(\log n)$.
- $\Rightarrow$ Total work is $\Theta(n^2 \log n)$.
- Why it works:
  - $\Theta(n^2)$ **common votes** produce linear-sized majority with constant probability.
  - $O(n^2/\log n)$ **extra votes** seen by one process change this enough to make a difference with probability $\ll 1/n$.

(Bracha and Rachman, WDAG 1991)

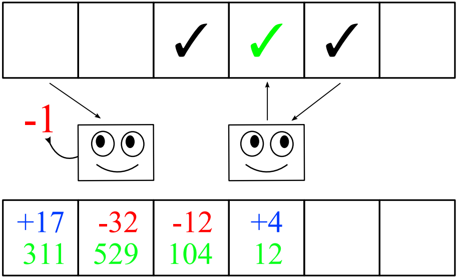
Attiya-Censor protocol



- Get all processes to agree on extra votes.
- $\Rightarrow$ OK to have $O(n^2)$ extra votes.
- $\Rightarrow$ Only need to check total every $O(n)$ votes.
- $\Rightarrow$ Amortized cost per vote $= O(1)$.
- $\Rightarrow$ Total cost $= O(n^2)$ (optimal).

Mechanism: *multi-writer* **termination bit** shuts down voting immediately as soon as one process sees enough votes.
(Attiya and Censor, JACM 2008)

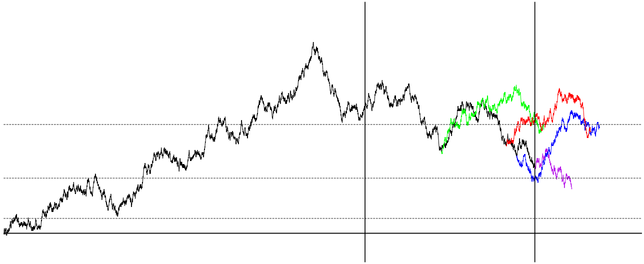
Getting rid of the multi-writer bit



Replace with randomized gossip:

- Each process has its own bit *done*$[i]$.
- Read uniformly chosen *done*$[r]$ before each vote.
- Stop and set my own *done* bit if I see somebody else is done.

Effect of done bits

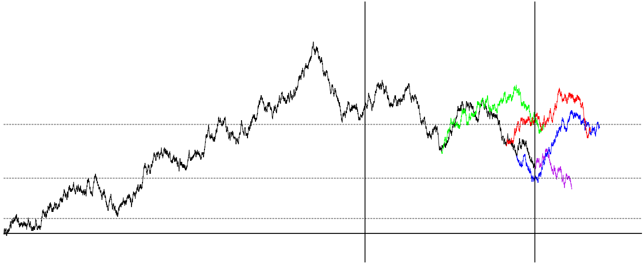


- If k *done* bits are set, $\Pr[\mathit{done}[r] = 1] = k/n$.
- $\Rightarrow$ on average, each process generates $\leq n/k$ more votes.
- $\Rightarrow$ on average, k-th process to set $\mathit{done}[i]$ sees $\leq n^2/k$ extra votes.
- We'll show stronger result that, with probability $1/2$, no process sees more than $2n^2/k$ extra votes.

$2n^2/k$ bound on extra votes

- Let **contribution** of a vote be number of *done* bits set when it is generated $\geq$ number of processes that include it in their extra votes.
- $Y_t = \sum(\text{contributions}) + n \cdot (\#\text{ of processes still voting})$.
- Each vote:
  - Raises left term by k.
  - Lowers right term by $n \cdot (k/n) = k$ on average.
  - Total effect is 0 on average.
- So $\mathrm{E}\left[\sum(\text{all contributions})\right] = \mathrm{E}[Y_\infty] \leq \mathrm{E}[Y_0] = n^2$.
- With prob. $1/2$, $\sum(\text{all contributions}) \leq 2n^2$.
- If I am k-th process to write *done*[i], extra votes I see all have contribution $\geq k$.
- $\Rightarrow$ I see $\leq 2n^2/k$ extra votes.

Full result



All of these events happen with constant probability:

- Total vote is more than $8n$ after $64n^2$ votes.
- Vote stays above $4n$ until all processes see $64n^2$.
- Extra votes don't push total below n:
  - $\Pr\left[X_k \leq -3n\right] \leq \exp\left(-\frac{(3n)^2}{2(2n^2/k)}\right) = \left(e^{-9/4}\right)^k$.
  - Sum is geometric series $< 1/8$.
- $\Rightarrow$ Everybody stays above $+n$.

Conclusions

- $O(n^2)$ total work for consensus with single-writer registers.
- Optimal even for multi-writer registers.
  (Attiya and Censor, JACM 2008)
- What about individual work?
  - Best known multi-writer bound is $O(n)$
    (Aspnes and Censor, SODA 2009).
  - Best known single-writer bound is $O(n \log^2 n)$
    (Aspnes and Waarts, SICOMP 1996).
  - Right answer is probably $O(n)$, but not clear how to get it.