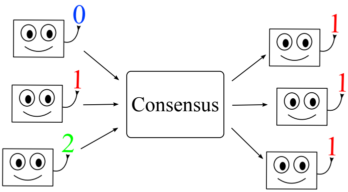
Faster randomized consensus with an oblivious adversary

James Aspnes
Yale

July 16th, 2012

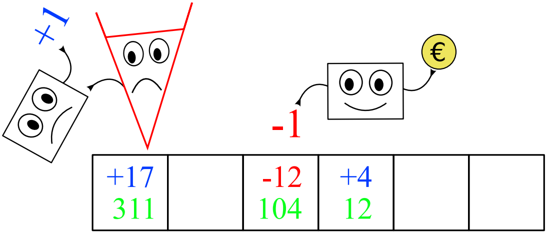
Consensus



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

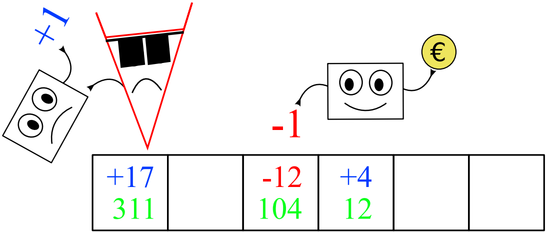
No deterministic solutions! (Fischer, Lynch, and Paterson 1985; Loui and Abu-Amara 1987)

Asynchronous shared-memory model



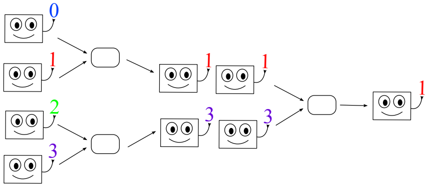
- n concurrent **processes** with **local coins**.
- Communication by reading and writing **atomic registers**.
- Timing controlled by an **adversary scheduler**.
- Algorithm is **wait-free**: tolerates $n-1$ **crash failures**.
- Cost measure: **expected individual steps**.

Oblivious adversary



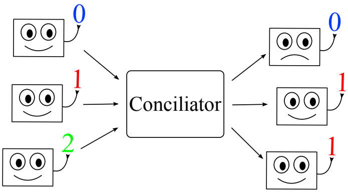
- Chooses schedule in advance.
- *Can* see algorithm.
- *Can't* see what algorithm does.
- Avoids $\Omega(n)$ lower bound for adaptive adversary due to Attiya and Censor (JACM 2008).

Previous results



- Long history of algorithms with $O(\log n)$ expected steps: (Aumann, PODC 1997; Aspnes, PODC 2010)
- Best lower bound is $\Omega(1)$ expected steps, from $\Omega(\log(1/\epsilon))$ steps to finish with probability at least $1 - \epsilon$. (Attiya and Censor-Hillel, SICOMP 2010).
- We'll show a new upper bound of $O(\log \log n)$.

Conciliators

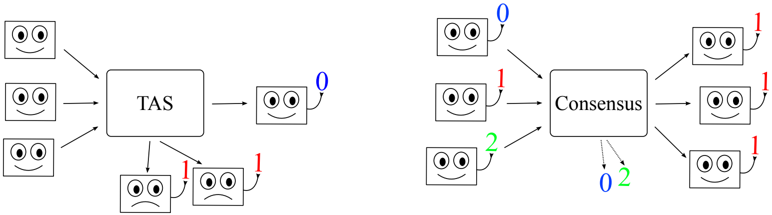


Monte Carlo version of consensus:

- **Termination:** All non-faulty processes terminate.
- **Validity:** Every output value is somebody's input.
- **Probabilistic agreement:** All output values are equal *with probability at least* δ.

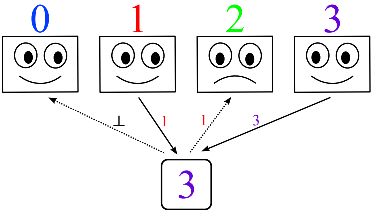
With m possible input values, can *detect* agreement (and get real consensus) with $O(\log m / \log \log m)$ overhead (Aspnes and Ellen, SPAA 2011).

Test-and-set



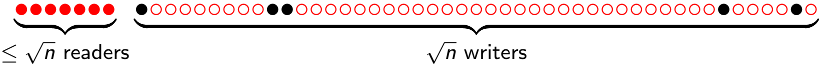
- Good randomized test-and-set implementations for oblivious-adversary model:
  - $O(\log \log n)$ (Alistarh and Aspnes, DISC 2011).
  - $O(\log^* n)$ (Giakkoupis and Woelfel, later in this session).
- Test-and-set gets *processes* to drop out.
- Consensus gets *values* to drop out.

Sifting processes for test-and-set



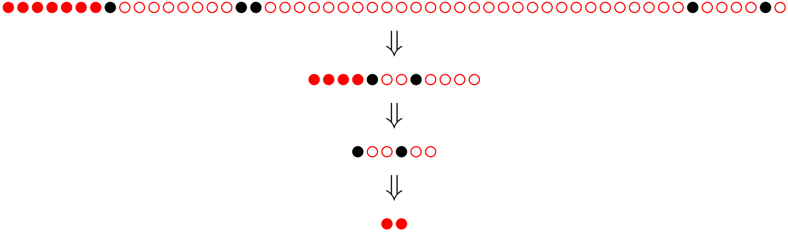
- Single multi-writer register, initially $\perp$.
- Each process **reads** with probability $1 - \frac{1}{\sqrt{n}}$, **writes** with probability $\frac{1}{\sqrt{n}}$.
- A process **survives** if it reads $\perp$ or writes.

Sifting: one round



- Because adversary is oblivious, coin-flips are independent of ordering.
- Before first write, all readers survive.
  - This is a waiting time process with expectation $\leq \frac{1}{p} = \sqrt{n}$.
- Otherwise, only writers survive.
  - $pn = \frac{1}{\sqrt{n}} \cdot n = \sqrt{n}$.
- Total expected survivors $\leq 2\sqrt{n}$.

Sifting: multiple rounds

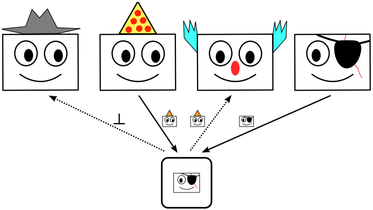


- Tune probabilities so that on average we go from k to $2\sqrt{k}$.
- Linearity of expectation gives

$$n,\ 2\sqrt{n},\ 2\sqrt{2\sqrt{n}},\ 2\sqrt{2\sqrt{2\sqrt{n}}},\ \ldots \leq 4n^{(1/2)^r}$$

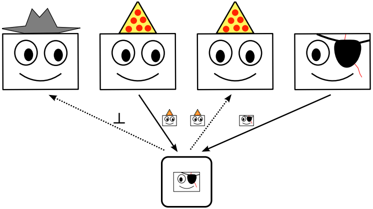
  expected survivors after r rounds.
- Converges to $O(1)$ expected survivors in $O(\log\log n)$ rounds.

Sifting personae for consensus



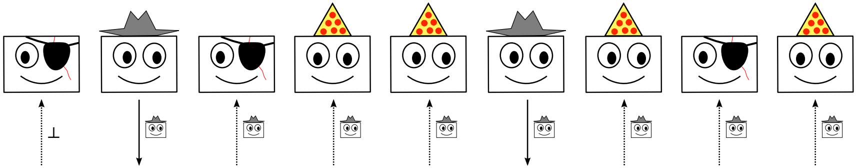
- Generate all coin-flips at start.
- Coin-flips + input = **persona**.
- When I write, I write my persona.
- When I read, I adopt any persona I see.

Sifting personae for consensus



- Generate all coin-flips at start.
- Coin-flips + input = **persona**.
- When I write, I write my persona.
- When I read, I adopt any persona I see.

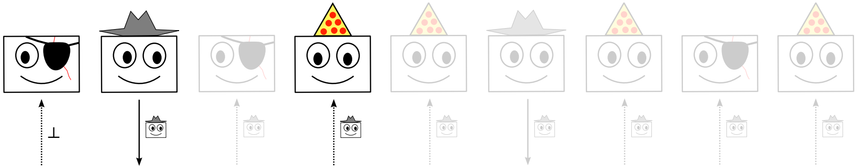
Sifting personae: analysis



- All processes with the same persona in some round do the same thing.
- If all copies write, persona survives (and maybe spreads to more processes) $\Rightarrow \sqrt{k}$ expected survivors.
- If they all read, at least one copy of persona survives if the first read sees $\perp$ (other copies might be overwritten) $\Rightarrow \leq \sqrt{k}$ more expected survivors

So average number of surviving personae is $2\sqrt{k}$, as in test-and-set.

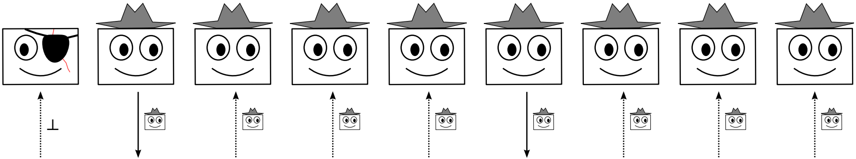
Sifting personae: analysis



- All processes with the same persona in some round do the same thing.
- If all copies write, persona survives (and maybe spreads to more processes) $\Rightarrow \sqrt{k}$ expected survivors.
- If they all read, at least one copy of persona survives if the first read sees $\perp$ (other copies might be overwritten) $\Rightarrow \leq \sqrt{k}$ more expected survivors

So average number of surviving personae is $2\sqrt{k}$, as in test-and-set.

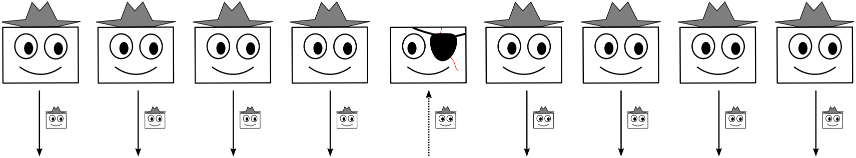
Sifting personae: analysis



- All processes with the same persona in some round do the same thing.
- If all copies write, persona survives (and maybe spreads to more processes) $\Rightarrow \sqrt{k}$ expected survivors.
- If they all read, at least one copy of persona survives if the first read sees $\perp$ (other copies might be overwritten) $\Rightarrow \leq \sqrt{k}$ more expected survivors

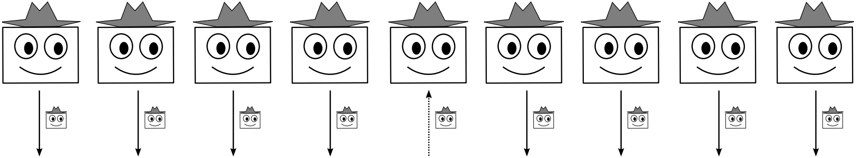
So average number of surviving personae is $2\sqrt{k}$, as in test-and-set.

Final stage



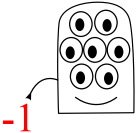
- After $O(\log\log n)$ rounds, switch to $\Pr[\text{write}] = 1/2$.
- This reduces expected surviving personae from $O(1)$ to $1 + \epsilon$ in $O(\log(1/\epsilon))$ additional rounds.
- Total cost to get $\Pr[\text{agree}] > 1 - \epsilon$ is $O(\log\log n + \log(1/\epsilon))$.
- Second term matches $\Omega(\log(1/\epsilon))$ lower bound of Attiya and Censor-Hillel (SICOMP 2010).

Final stage



- After $O(\log\log n)$ rounds, switch to $\Pr[\text{write}] = 1/2$.
- This reduces expected surviving personae from $O(1)$ to $1 + \epsilon$ in $O(\log(1/\epsilon))$ additional rounds.
- Total cost to get $\Pr[\text{agree}] > 1 - \epsilon$ is $O(\log\log n + \log(1/\epsilon))$.
- Second term matches $\Omega(\log(1/\epsilon))$ lower bound of Attiya and Censor-Hillel (SICOMP 2010).

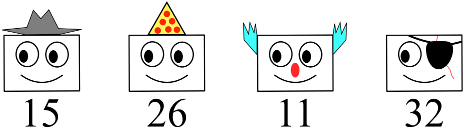
Cheap snapshots



+17 311		-12 104	+4 12		

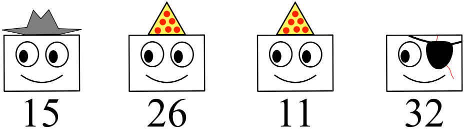
- **Snapshot** operation reads all registers simultaneously.
- In the **cheap snapshot** model, this costs 1 operation.
- Model for Attiya+Censor-Hillel weak-adversary lower bound.
- Also popular with topologists.
- We'll show that this gives consensus in $O(\log^* n)$ expected operations.

Consensus with cheap snapshots



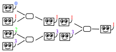
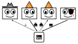
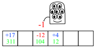
- Persona now is input plus random **priority** for each round.
- Algorithm for one round:
  - Write my persona to my own register.
  - Take snapshot and adopt highest-priority persona I see.
- Pr[i-th persona to be written survives] $\leq (1/i)$.
- So in one round, expected survivors goes from k to $\sum_{i=1}^{k}(1/i) = O(\log k)$.
- Repeat $O(\log^* n)$ times on average to get to 1.

Consensus with cheap snapshots



- Persona now is input plus random **priority** for each round.
- Algorithm for one round:
  - Write my persona to my own register.
  - Take snapshot and adopt highest-priority persona I see.
- Pr[i-th persona to be written survives] $\leq (1/i)$.
- So in one round, expected survivors goes from k to $\sum_{i=1}^{k}(1/i) = O(\log k)$.
- Repeat $O(\log^* n)$ times on average to get to 1.

Conclusions

	$O(\log n)$	previous bounds
	$O(\log \log n)$	new bound for multi-writer registers
	$O(\log^* n)$	new bound for cheap snapshots
	$\Omega(1)$	best known lower bound

- Conciliator algorithms work for arbitrarily many inputs m, but detecting agreement takes $O(\log m / \log \log m)$ steps, which dominates $O(\log \log n)$ unless m is small.
- Cheap-snapshot bound shows that combining local coins isn't the hard part.
- Maybe we can get $O(1)$?