The Triumph of Randomization

The Big Picture

- Does randomization make for more powerful algorithms?
  - Does randomization expand the class of problems solvable in polynomial time?
  - Does randomization help compute problems fast in parallel in the PRAM model?

You tell me!

The Triumph of Randomization?

Well, at least for distributed computations!

- no deterministic 1-crash-resilient solution to Consensus
- f-resilient randomized solution to consensus ($f < n/2$) for crash failures
- randomized solution for Consensus exists even for Byzantine failures!

A simple randomized algorithm

M. Ben Or. "Another advantage of free choice: completely asynchronous agreement protocols" (PODC 1983, pp. 27-30)

- exponential number of operations per process
- BUT more practical protocols exist
  - down to $O(n\, log^2 n)$ expected operations/process
  - $n-1$ resilient

The protocol's structure

An infinite repetition of asynchronous rounds

- in round r, p only handles messages with timestamp r
- each round has two phases
- in the first, each p broadcasts an a-value which is a function of the b-values collected in the previous round (the first a-value is the input bit)
- in the second, each p broadcasts a b-value which is a function of the collected a-values
- decide stutters

Ben Or's Algorithm

a_p:= input bit; r:= 1;
repeat forever
{phase 1}
send (a_p, r) to all
Let A be the multiset of the first $n-f$ a-values with timestamp r received
if $(\exists v \in \{0,1\} : \forall a \in A : a = v)$ then b_p := v
else b_p := $\perp$
{phase 2}
send (b_p, r) to all
Let B be the multiset of the first $n-f$ b-values with timestamp r received
if $(\exists v \in \{0,1\} : \forall b \in B : b = v)$ then decide(v); a_p:= v
else if $(\exists b \in B : b \neq \perp)$ then a_p:= b
else a_p:= $\$$ {$\$$ is chosen uniformly at random to be 0 or 1}
r := $r+1$

Validity

a_p := input bit; r:= 1;
repeat forever
{phase 1}
send (a_p, r) to all
Let A be the multiset of the first $n-f$ a-values with timestamp r received
if $(\exists v \in \{0,1\} : \forall a \in A : a = v)$ then b_p := v
else b_p := $\perp$
{phase 2}
send (b_p, r) to all
Let B be the multiset of the first $n-f$ b-values with timestamp r received
if $(\exists v \in \{0,1\} : \forall b \in B : b = v)$ then decide(v); a_p:= v
else if $(\exists b \in B : b \neq \perp)$ then a_p:= b
else a_p := $\$$ {$\$$ is chosen uniformly at random to be 0 or 1}
r := $r+1$

Validity

a_p := input bit; r:= 1;
repeat forever
{phase 1}
send (a_p, r) to all
Let A be the multiset of the first $n-f$ a-values with timestamp r received
if $(\exists v \in \{0,1\} : \forall a \in A : a = v)$ then b_p := v
else b_p := $\perp$
{phase 2}
send (b_p, r) to all
Let B be the multiset of the first $n-f$ b-values with timestamp r received
if $(\exists v \in \{0,1\} : \forall b \in B : b = v)$ then decide(v); a_p:= v
else if $(\exists b \in B : b \neq \perp)$ then a_p:= b
else a_p := $\$$ {$\$$ is chosen uniformly at random to be 0 or 1}
r := $r+1$

- All identical inputs (i)
- Each process set a-value := i and broadcasts it to all
- Since at most f faulty, every correct process receives at least $n-f$ identical a-values in round 1
- Every correct process sets b-value := i and broadcasts it to all
- Again, every correct process receives at least $n-f$ identical i b-values in round 1 and decides

A useful observation

```
1:  a_p := input bit;  r := 1;
2:  repeat forever
3:  {phase 1}
4:  send (a_p, r) to all
5   Let A be the multiset of the first n−f a-values with
       timestamp r received
6:  if  (∃v ∈ {0,1} : ∀a ∈ A : a = v) then b_p := v
7:  else  b_p := ⊥
8:  {phase 2}
9:  send (b_p, r) to all
10: Let B be the multiset of the first n−f b-values with
       timestamp r received
11: if (∃v ∈ {0,1} : ∀b ∈ B : b = v)  then decide(v);   a_p := v
12: else if (∃b ∈ B : b ≠ ⊥)  then  a_p :=  b
13: else  a_p := $   {$ is chosen uniformly at random
                        to be 0 or 1}
14:  r := r+1
```

Lemma For all r, either

$b_{p,r} \in \{1, \perp\}$ for all p or

$b_{p,r} \in \{0, \perp\}$ for all p

A useful observation

```
1:  a_p := input bit;  r := 1;
2:  repeat forever
3:  {phase 1}
4:  send (a_p, r) to all
5   Let A be the multiset of the first n−f a-values with
       timestamp r received
6:  if  (∃v ∈ {0,1} : ∀a ∈ A : a = v) then b_p := v
7:  else  b_p := ⊥
8:  {phase 2}
9:  send (b_p, r) to all
10: Let B be the multiset of the first n−f b-values with
       timestamp r received
11: if (∃v ∈ {0,1} : ∀b ∈ B : b = v)  then decide(v);   a_p := v
12: else if (∃b ∈ B : b ≠ ⊥)  then  a_p :=  b
13: else  a_p := $   {$ is chosen uniformly at random
                        to be 0 or 1}
14:  r := r+1
```

Lemma For all r, either

$b_{p,r} \in \{1, \perp\}$ for all p or

$b_{p,r} \in \{0, \perp\}$ for all p

Proof By contradiction.

Suppose p and q at round r such that $b_{p,r}$ = 0 and $b_{q,r}$ = 1

From lines 6,7 p received $n-f$ distinct 0s, q received $n-f$ distinct 1s.

Then, $2(n-f) \leq n$, implying $n \leq 2f$

Contradiction

Corollary It is impossible that two processes p and q decide on different values at round r

Agreement


```
1:  a_p := input bit;  r := 1;
2:  repeat forever
3:  {phase 1}
4:  send (a_p, r) to all
5   Let A be the multiset of the first n−f a-values with
       timestamp r received
6:  if  (∃v ∈ {0,1} : ∀a ∈ A : a = v) then b_p := v
7:  else  b_p := ⊥
8:  {phase 2}
9:  send (b_p, r) to all
10: Let B be the multiset of the first n−f b-values with
       timestamp r received
11: if (∃v ∈ {0,1} : ∀b ∈ B : b = v)  then decide(v);   a_p := v
12: else if (∃b ∈ B : b ≠ ⊥)  then  a_p :=  b
13: else  a_p := $   {$ is chosen uniformly at random
                        to be 0 or 1}
14:  r := r+1
```


- Let r be the first round in which a decision is made
- Let p be a process that decides in r

Agreement

```
1:  a_p := input bit;  r := 1;
2:  repeat forever
3:  {phase 1}
4:  send (a_p, r) to all
5   Let A be the multiset of the first n−f a-values with
       timestamp r received
6:  if  (∃v ∈ {0,1} : ∀a ∈ A : a = v) then b_p := v
7:  else  b_p := ⊥
8:  {phase 2}
9:  send (b_p, r) to all
10: Let B be the multiset of the first n−f b-values with
       timestamp r received
11: if (∃v ∈ {0,1} : ∀b ∈ B : b = v)  then decide(v);   a_p := v
12: else if (∃b ∈ B : b ≠ ⊥)  then  a_p :=  b
13: else  a_p := $   {$ is chosen uniformly at random
                        to be 0 or 1}
14:  r := r+1
```

- Let r be the first round in which a decision is made
- Let p be a process that decides in r

- By the Corollary, no other process can decide on a different value in r
- To decide, p must have received $n-f$ “i” from distinct processes
- every other correct process has received “i” from at least $n-2f \geq 1$
- By lines 11 and 12, every correct process sets its new a-value to for round $r+1$ to “i”
- By the same argument used to prove Validity, every correct process that has not decided “i” in round r will do so by the end of round $r+1$

Termination I

```
1:  a_p := input bit;  r := 1;
2:  repeat forever
3:  {phase 1}
4:  send (a_p, r) to all
5   Let A be the multiset of the first n−f a-values with
        timestamp r received
6:  if (∃v ∈ {0,1} : ∀a ∈ A : a = v) then b_p := v
7:  else b_p := ⊥
8:  {phase 2}
9:  send (b_p, r) to all
10: Let B be the multiset of the first n−f b-values with
        timestamp r received
11: if (∃v ∈ {0,1} : ∀b ∈ B : b = v) then decide(v);  a_p := v
12: else if (∃b ∈ B : b ≠ ⊥) then  a_p := b
13: else  a_p := $   {$ is chosen uniformly at random
                      to be 0 or 1}
14:  r := r+1
```

- Remember that by Validity, if all (correct) processes propose the same value "i" in phase 1 of round r, then every correct process decides "i" in round r.
- The probability of all processes proposing the same input value (a landslide) in round 1 is

  Pr[landslide in round 1] = $1/2^n$
- What can we say about the following rounds?

Termination II

```
1:  a_p := input bit;  r := 1;
2:  repeat forever
3:  {phase 1}
4:  send (a_p, r) to all
5   Let A be the multiset of the first n−f a-values with
        timestamp r received
6:  if (∃v ∈ {0,1} : ∀a ∈ A : a = v) then b_p := v
7:  else b_p := ⊥
8:  {phase 2}
9:  send (b_p, r) to all
10: Let B be the multiset of the first n−f b-values with
        timestamp r received
11: if (∃v ∈ {0,1} : ∀b ∈ B : b = v) then decide(v);  a_p := v
12: else if (∃b ∈ B : b ≠ ⊥) then  a_p := b
13: else  a_p := $   {$ is chosen uniformly at random
                      to be 0 or 1}
14:  r := r+1
```

- In round r > 1, the a-values are not necessarily chosen at random!
- By line 12, some process may set its a-value to a non-random value v
- By the Lemma, however, all non-random values are identical!
- Therefore, in every r there is a positive probability (at least $1/2^n$) for a landslide
- Hence, for any round r

  Pr[no lanslide at round r] $\leq 1 - 1/2^n$
- Since coin flips are independent:

  Pr[no lanslide for first k rounds] $\leq (1-1/2^n)^k$
- When $k = 2^n$ this value is about 1/e; then, if $k = c2^n$

  Pr[landslide within k rounds] ≥

  $$1-(1-1/2^n)^k \approx 1 - 1/e^c$$

  which converges quickly to 1 as c grows