Distributed Systems

Q&A-Session

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Pre-determined information

- King & Queen algorithm: how are kings and queens pre-determined among nodes, i.e. doesn't this imply "consensus before consensus"?
  - e.g. Every process/server has a unique id. In the first round the process with id=1 is the king, in the second round...
  - Just having an id is no consensus.

Synchronous rounds

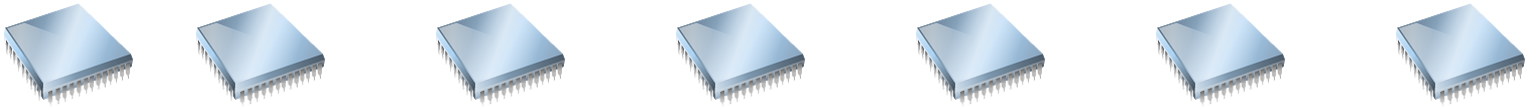
- What is a "synchronous round", i.e. how do processes keep their rounds in sync? (for example in king algorithm)
  - Theory
    - just accept the model, it works because we say it works.
  - Practice
    - e.g. a special “clock” server that can be accessed by everyone.
    - Local clocks, synchronized every time some servers communicate with each other.
    - Using a protocol like Lamport time.

Majority vs. Quorum

- How is PBFT's quorum different from Paxos' majority set?
  - Majority Set: Intersection contains at least one server.
    - Crashed servers do not count
    - Size approximately ½ of all servers
  - Quorum: Intersection contains at least one correct server.
    - Crashed servers do not count
    - Size approx. ½ of correct servers + #Byzantine servers

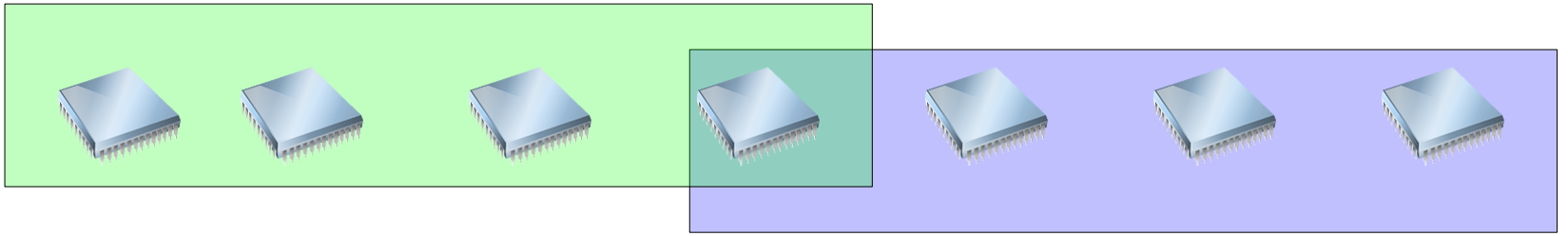
Majority vs. Quorum

- How is PBFT's quorum different from Paxos' majority set?



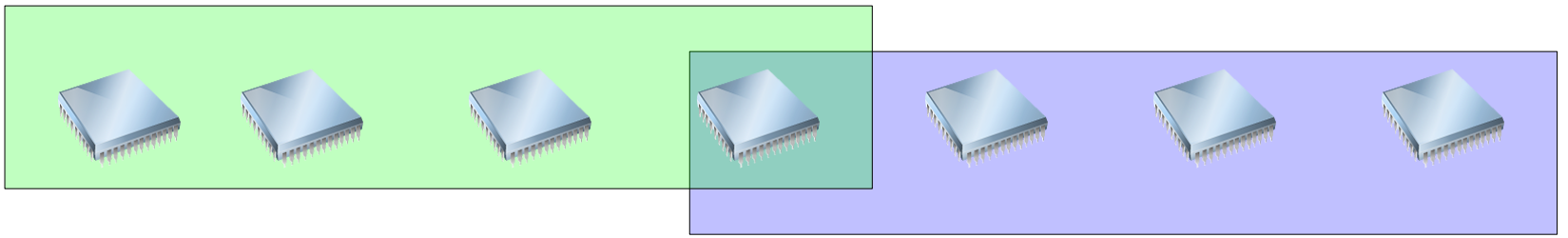
Majority vs. Quorum

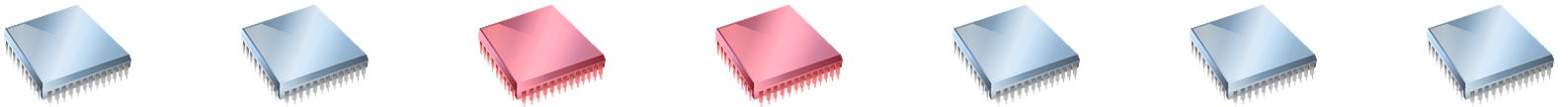
- How is PBFT's quorum different from Paxos' majority set?



Majority vs. Quorum

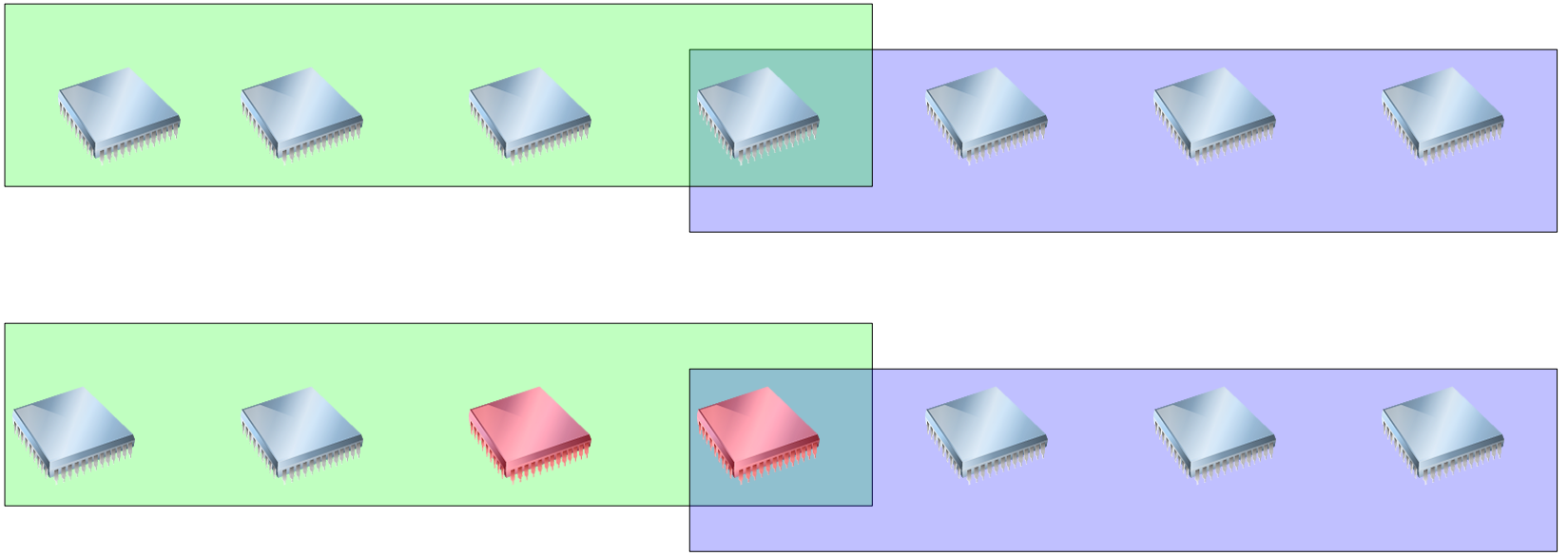
- How is PBFT's quorum different from Paxos' majority set?





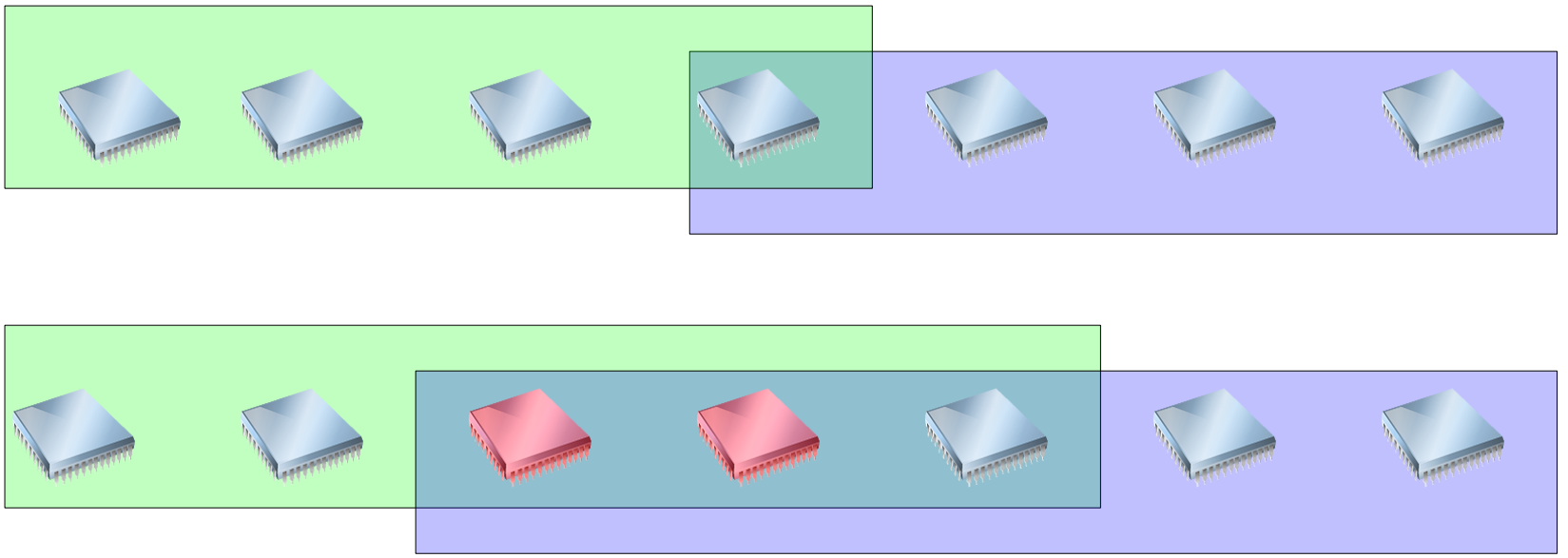
Majority vs. Quorum

- How is PBFT's quorum different from Paxos' majority set?



Majority vs. Quorum

- How is PBFT's quorum different from Paxos' majority set?



Majority vs. Quorum

- How is PBFT's quorum different from Paxos' majority set?
  - Majority Set: 20 servers, 5 of them may crash
    - Majority is 11
  - Quorum: 20 servers, 5 of them Byzantine
    - Quorum is 13
    - 12 would be not enough:
      - 5 Byzantine + 7 correct vote 0
      - 5 Byzantine + 7 correct vote 1

Randomized Algorithm

- What is the intuition behind choosing n-4f as the condition in the else-if branch on slide 6/137?
  - The algorithm would work with other numbers. It's more or less a random choice.
  - The numbers in the proofs would be different if it were not n-4f
  - And there are upper and lower boundaries for a useful condition.

Randomized Algorithm

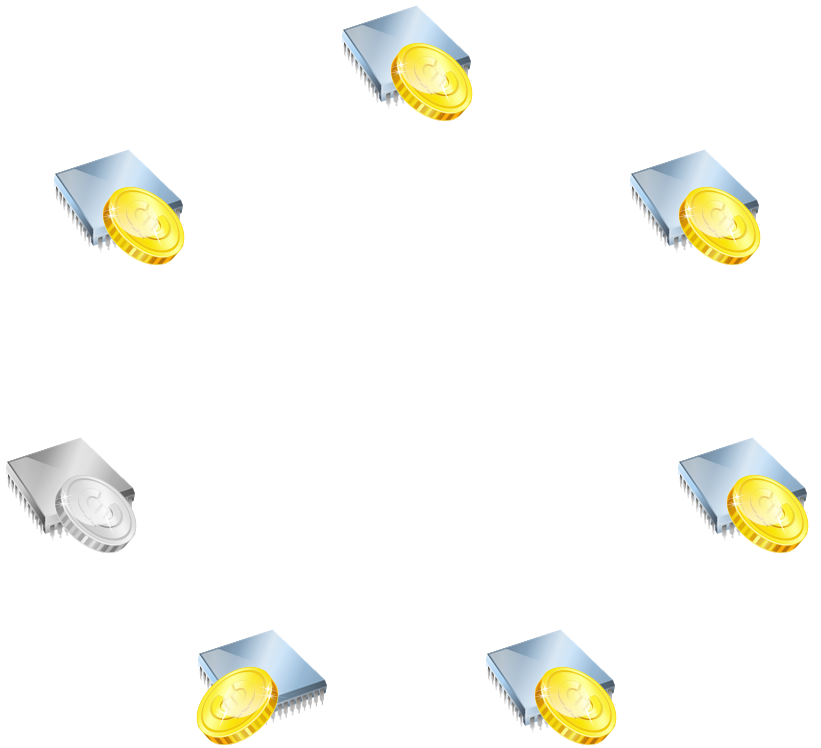
- Slide 6/138: why n-3f processes, not n-2f? A process decides on x once it received n-2f proposals for some value.
  - n-2f processes - f Byzantine processes
  - ... = n-3f correct processes

Randomized Algorithm

- Why do the other process receive x at least n-4f times?
  - There was some discussion, this one is the correct solution:
  - We know n-3f correct processes must have sent x. Because a process waits until it receives n-f values, it may stop listening once it received 4n-f times x.
  - Example: first 3f processes send y, then n-3f processes send x. The last f x's are not received because a process stops waiting once it got n-f values.

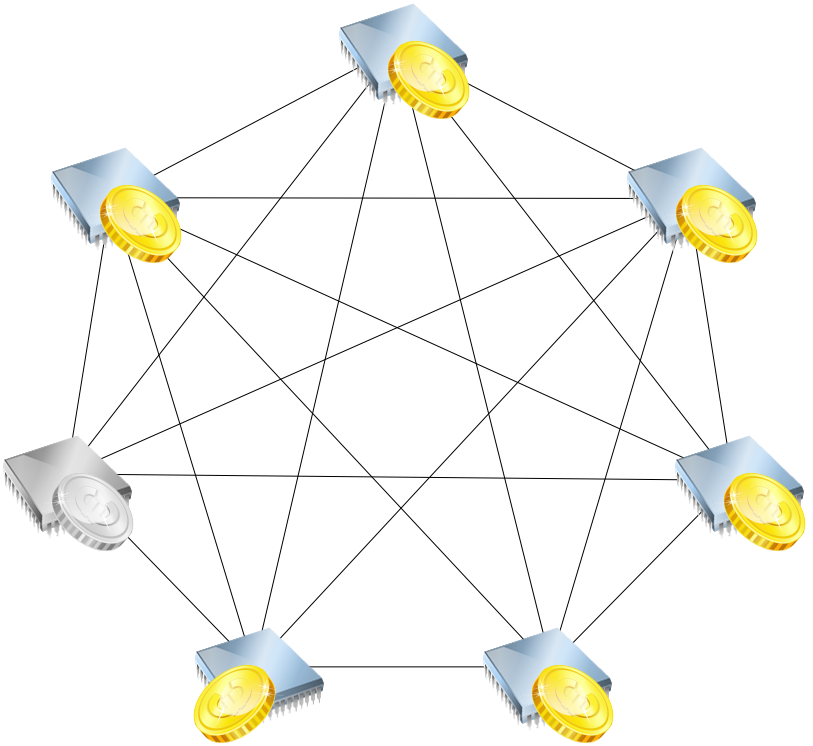
Shared Coin

- How does the Shared Coin Algorithm work?
  - Local coin = 0 with probability $\left(\frac{1}{n}\right)$, 1 with p = $\left(1-\frac{1}{n}\right)$



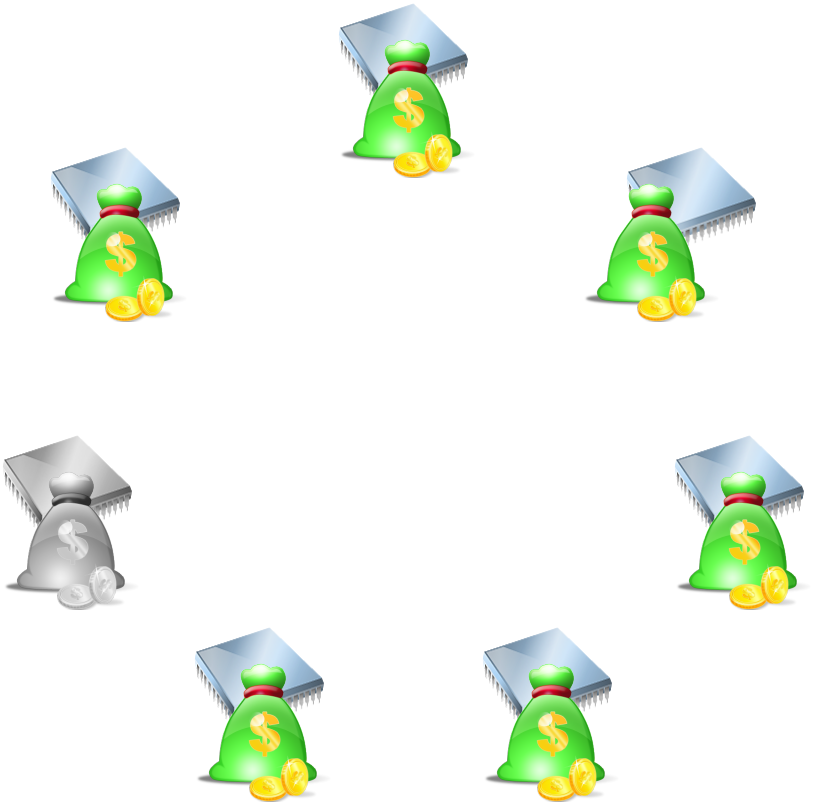
Shared Coin

- How does the Shared Coin Algorithm work?
  - Broadcast local coin



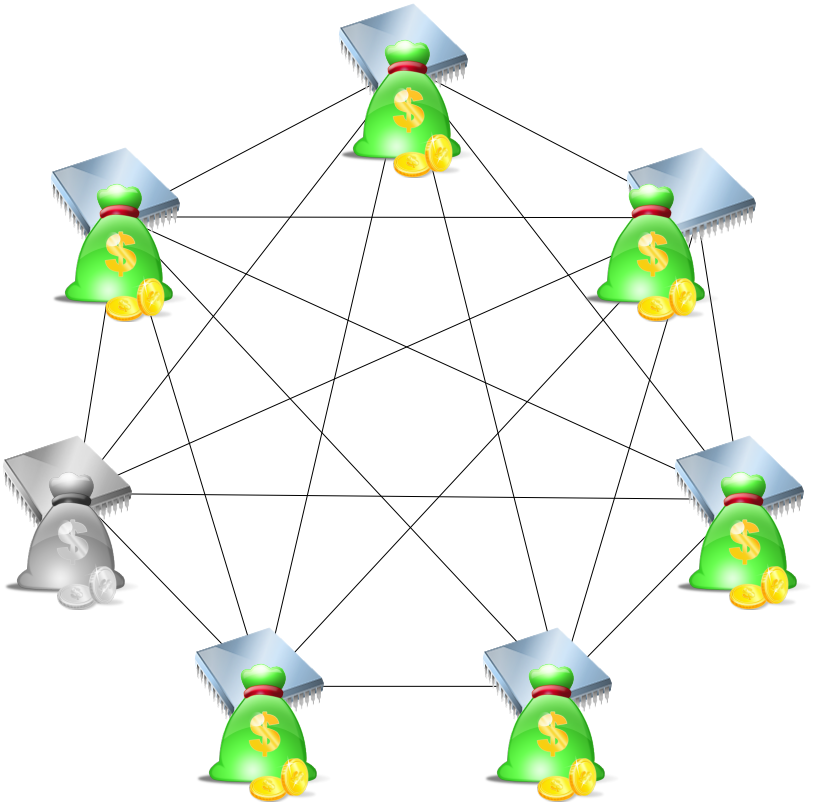
Shared Coin

- How does the Shared Coin Algorithm work?
  - Collect n-f coins



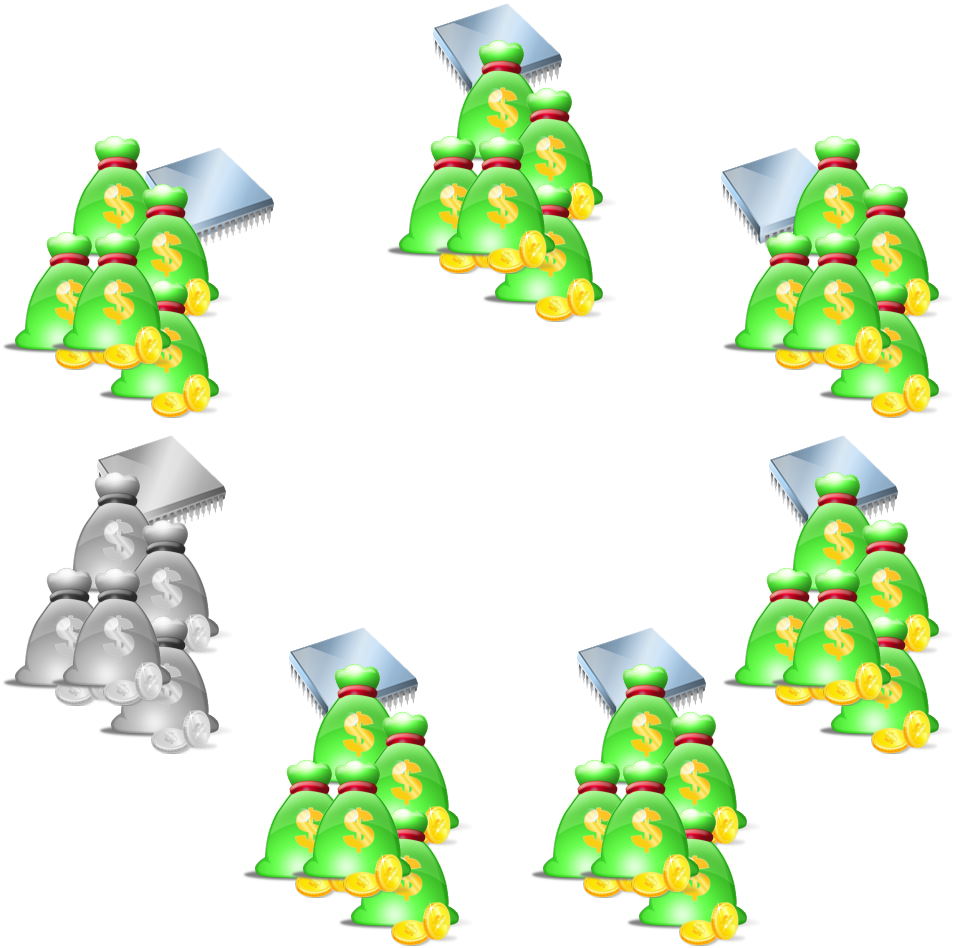
Shared Coin

- How does the Shared Coin Algorithm work?
  - Broadcast coin sets



Shared Coin

- How does the Shared Coin Algorithm work?
  - Collect n-f coin sets



Shared Coin

- How does the Shared Coin Algorithm work?
  - If all coins are 1: decide 1
  - Else: decide 0

Shared Coin

- How does the Shared Coin Algorithm work?
  - p(all processes decide 0) = 0.37
  - p(all processes decide 1) = 0.28
  - p(decision is inconclusive) = 0.35

  - Not a consensus algorithm itself. Needs some additional algorithm to check the result!
  - Can be used to generate the input of a randomized algorithm.

Shared Coin

- What's up with this “at least 1/3 of all coins are seen by everybody”?
  - Required to calculate the probability that all processes decide on 0.
  - for f+1 coins (=1/3 of all coins): each coin is in f+1 coin sets (=one set is certainly known to all processes)
  - Slide 6/143: the faulty processes 2 and 4 are not part of the matrix!

Shared Coin

- Shared coin analysis, proof (slide 6/145) - how is the probability derived? in particular 1/e?
  - coin c_i = 0 with probability $\left(\frac{1}{n}\right)$
  - coin c_i = 1 with probability $\left(1-\frac{1}{n}\right)$
  - n coins = 1: $f(n)=\left(1-\frac{1}{n}\right)^n$
  - $\ln f(n)=n\ln\left(1-\frac{1}{n}\right)$

    $\lim_{n\to\infty}\ln f(n)=\lim_{n\to\infty} n\ln\left(1-\frac{1}{n}\right)=-1$ *by L'Hopistal's rule*

    $\ln\lim_{n\to\infty} f(n)=-1$

    $\lim_{n\to\infty} f(n)=\frac{1}{e}$

Shared Coin

- Shared coin analysis, proof (slide 6/145) - how is the probability derived? in particular 1/e?
  - at least n/3 coins are seen (6/142-145):
    - remember, a 'coin' is a value: '0' or '1'
  - at least one of them is 0

$$\underbrace{\underbrace{\underbrace{1-\left(1-\underbrace{\frac{1}{n}}_{p(coin=0)}\right)}_{p(coin=1)}}_{p\left(\frac{n}{3}\,coins=1\right)}}_{p(at\ least\ one\ is\ not\ 1)}^{\frac{n}{3}} \approx 1-\left(\frac{1}{e}\right)^{\frac{1}{3}}$$

PBFT

- Why are 2f accepted prepare messages enough to multicast a commit? Why not 2f+1 as is required for responding?
  - This is said on slide 7/74, however on slide 7/78 the number changes to 2f+1.
  - This is just an error on the slides.

Chubby

- Slide 7/56: System increases lease times from 12s up to 60s under heavy load

  Why?

  - Work needs to be done at the beginning and the end of a lease.
  - Longer lease times means less beginnings and endings.
  - .. and that means less work

Chubby

- Slide 7/54: Lock Holder Crash Heristic I: is this sequencer included in all requests the client is doing?
  - No, the client can decide when to send the sequencer and when not. There are even safe guards against servers and clients which do not support sequencers at all.

What to learn?

- Is it necessary to know by heart the various consensus protocols like King, Queen, Paxos, or PBFT, Zyzzyva?
  - Given an picture (like 7/91) or pseudo code:
    - explain what happens (in details)
    - do calculations, explain how the algorithm reacts to modifications.
  - Given nothing
    - explain basic idea
    - compare with other protocols

What to learn?

- Distributed Hashtables:
  - Do not learn slide 8/121 and everything after 8/121!

Calculations

- Calculations in P2P/DHT, in particular how to best approach a proof for O(log n) for search using skip lists, butterfly, de bruijn graphs?
  - Show #results decreases by ½ in each step.
  - Map data structure to search tree
  - Really depends on the concrete situation...

  - The exam is not about complex mathematics

How is the grade calculated?

- Exam: 50% of the grade comes from Prof. Wattenhofers part, 50% of the grades depend on Prof. Matterns part.
- After todays remark that Matters part should count more, because it is longer, we are going to discus whether the practical exercises should count 15% of the exam (hence 7.5% of the final grade) or 15% of the final grade.

  We will publish the decision on the website.