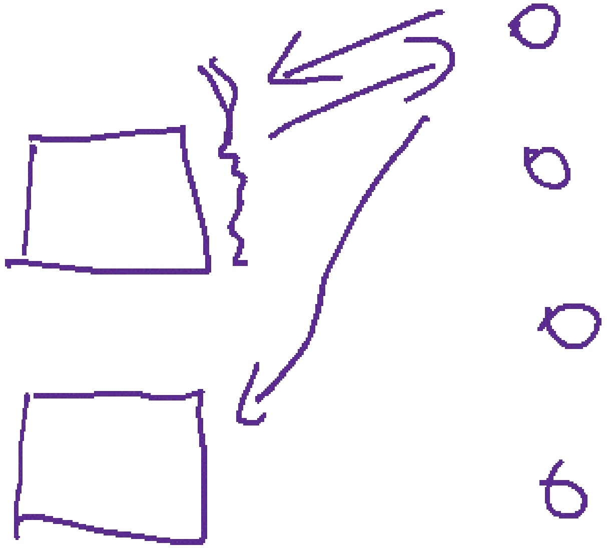
Distributed Computing Informatyka algrytmiczna 2021

Prof. Mirosław Kutyłowski

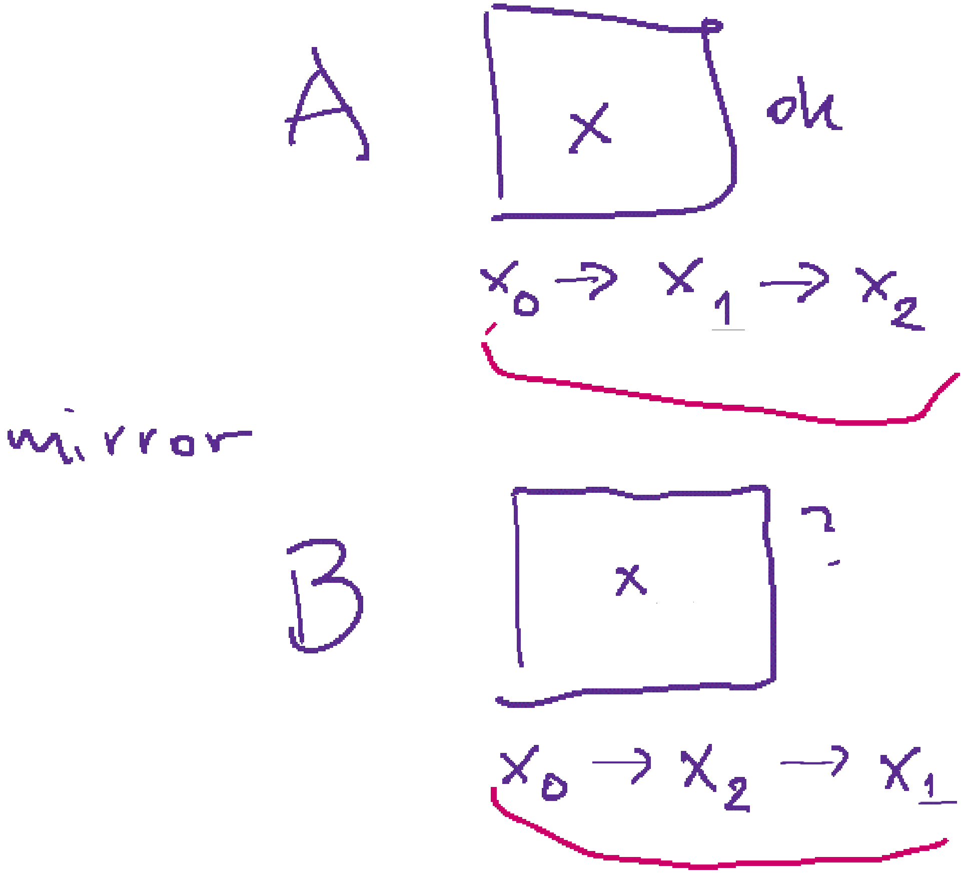
9: Quorum systems

Quorum problem:

- many clients
- *n* mirror servers holding the client's values
- communication is not always reliable
- some servers might be down



Problem



$\underbrace{x_0 \rightarrow x_1 \rightarrow x_2}$

$\underbrace{x_0 \rightarrow x_2 \rightarrow x_1}$

WARUNEK:

WRITE ok if

na każdym serwerze ok

1) czekać ☹

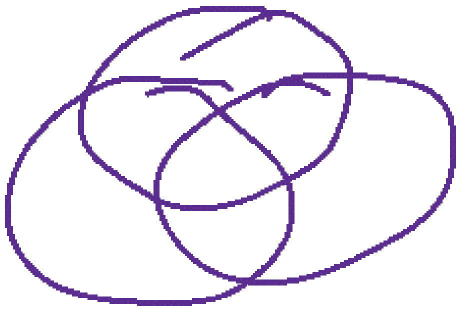
2) ignorować ☺ !

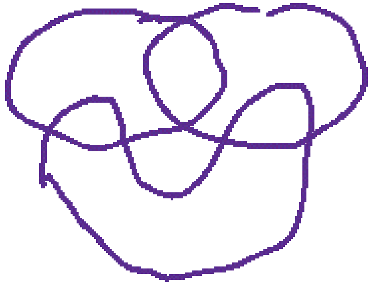
Quorum system

Q is a quorum system on a set of servers S iff

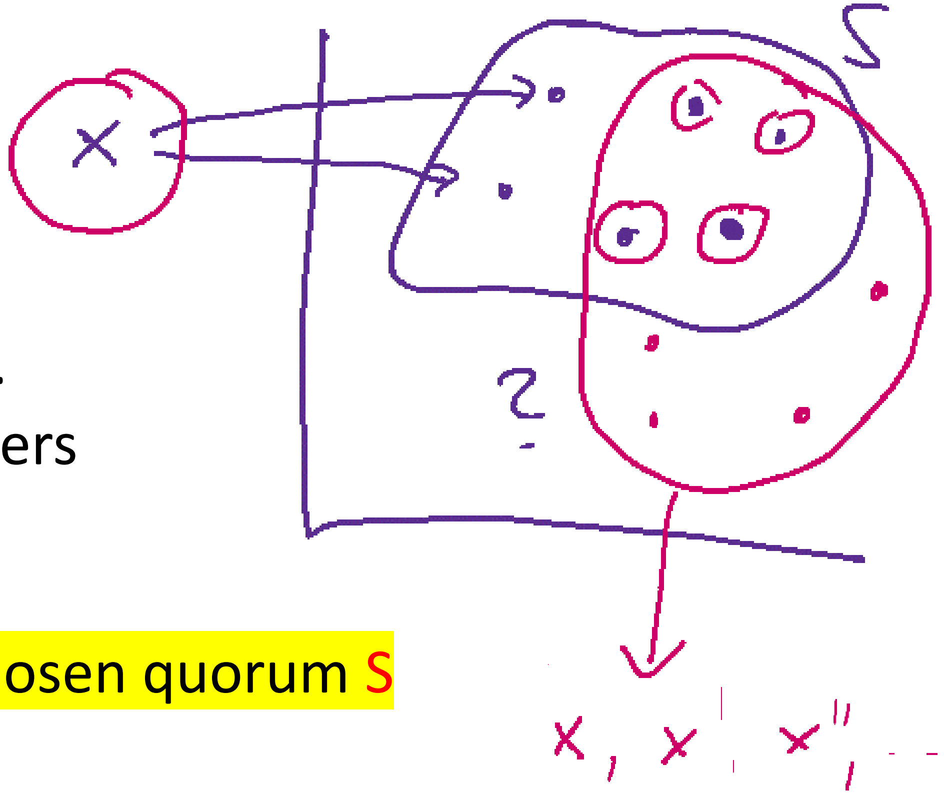
for any quorum set U_1, U_2 from Q, the intersection of U_1, U_2 is nonempty

zbiór serwerów





Using a quorum



- A server might be (temporarily) offline, delayed, …
- A client keeps the value of variable x on all all servers

- WRITE operation:
  - an update request sent to all servers from a chosen quorum S
  - the client updates x on every server form S
  - update sequential number stored as well

- READ operation:
  - A read request sent to all servers from a chosen quorum S’
  - a reader fetches the value of x from every server form S’
  - the most recent value accepted

Why it works

Due to the quorum property:

- if the most recent update of x stored at quorum set U_1,

and a reader A fetches the data from a quorum set U_2,

- then A will learn the most recent value of x

(simply, U_1 and U_2 intersect!)

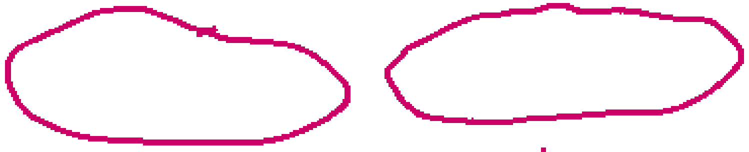
Example: majority quorums

Each set of more than n/2 servers is a quorum

$n = \cancel{1}, \cancel{2}, 3, 4, \ldots$

4 z 7

podzbiór $> \frac{n}{2}$ elementów



$$\underbrace{\frac{n}{2}+\varepsilon \qquad \frac{n}{2}+\delta}$$

$$n+\varepsilon+\delta > n$$

$Q_1 \quad p_1'$

$Q_2 \quad p_2'$

$Q_3 \quad p_3'$

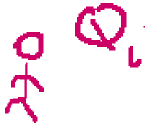
$S_1 \in Q_1 \to p_1$

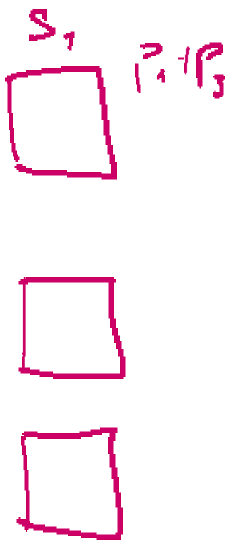
$S_1 \notin Q_2 \to 0$

$S_1 \in Q_3 \to p_3$

Complexity measures:

Strategy: probability distribution over the choice of a quorum from the quorum set





Load:

for server v and strategy Z : probability to serve a request when quorum chosen according to strategy Z

for the system: maximum over all servers

for quorum system: the system load for the best strategy **Z**

Work:

$p_1 \cdot |Q_1| + p_2 \cdot |Q_2| + p_3 \cdot |Q_3|$

for a quorum U: cardinality of U

for strategy Z: the expected size of the quorum chosen to execute an operation

for quorum system: work for the best strategy

Majority quorum

$\frac{n}{2} + \varepsilon = m \qquad m - 1 \leq \frac{n}{2}$

Load:

for server **v** **and strategy** **Z** **:** probability to serve a request when quorum chosen according to strategy Z $\frac{m}{n}$

for the system: maximum over all servers $\frac{m}{n}$

for quorum system: the system load for the best strategy **Z**

$\text{load} \approx \frac{1}{2}$

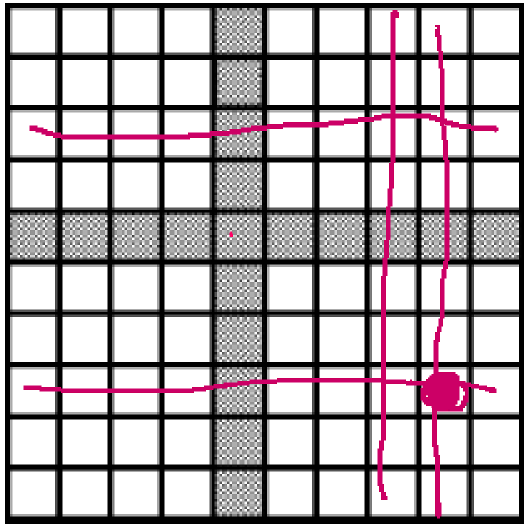
Majority quorum

Work:

- **for a quorum U:** cardinality of U m
- **for strategy Z:** the expected size of the quorum chosen to execute an operation m
- **for quorum system:** work for the best strategy m

$$\text{work} \approx \frac{n}{2}$$

Grid quorum system



A quorum set: a row + a column

Uniform strategy:

Load: 2/SQRT(n)- 1/n $\approx \frac{2}{\sqrt{n}}$

Work 2*SQRT(n)-1 $\approx 2\sqrt{n}$

MAJ.

2

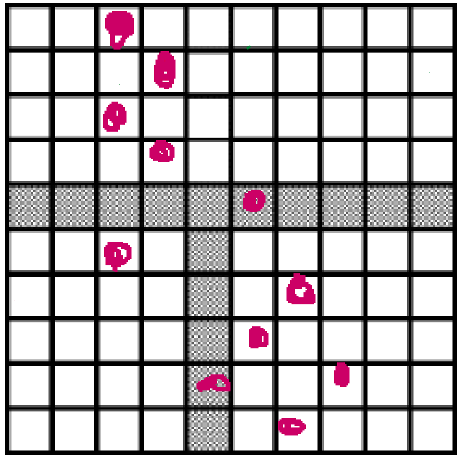
$\frac{n}{2}$

[illegible]

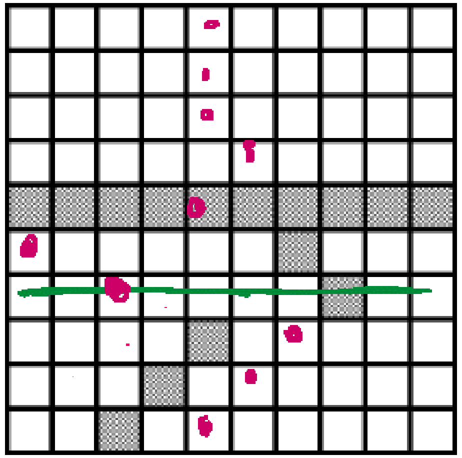
$\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{n}} - \frac{1}{n}$

Other grid quorum system

(a)



(b)



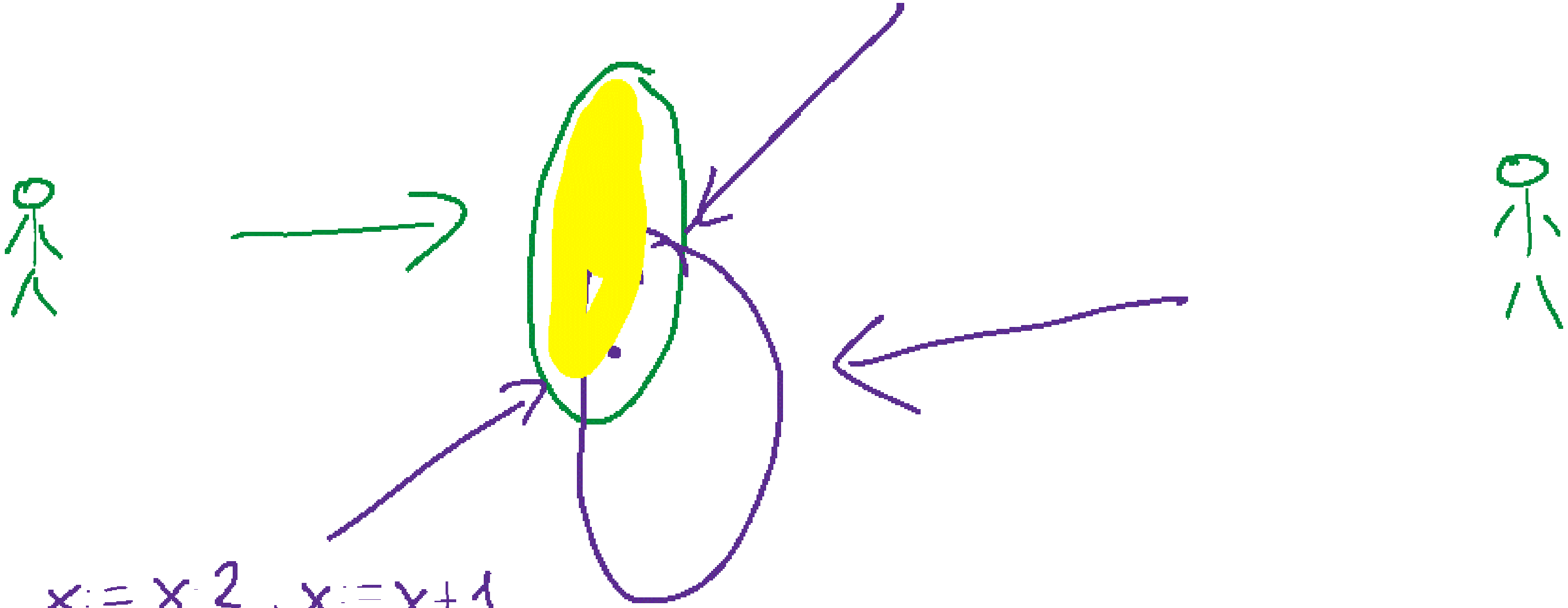
- Now two quorums intersect on one point
- Load and work are slightly reduced

How to coordinate the operations of multiple clients

- ~~first lock a quorum~~
- then execute

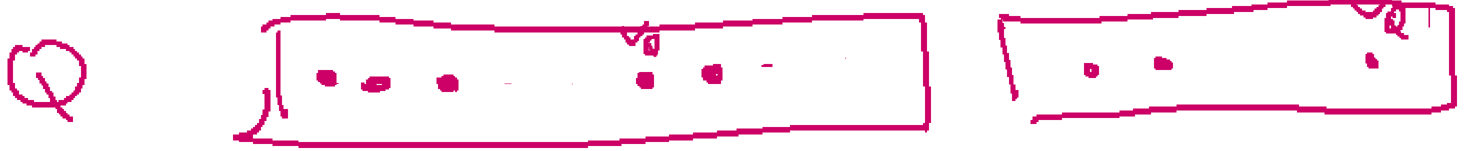
$0 \to \underline{1} \to 2$

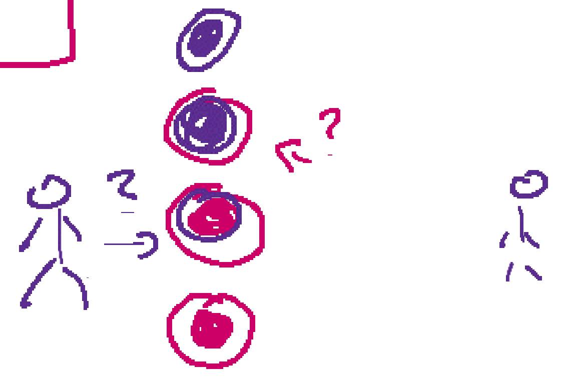
$x := x+1,\ x := x \cdot 2$



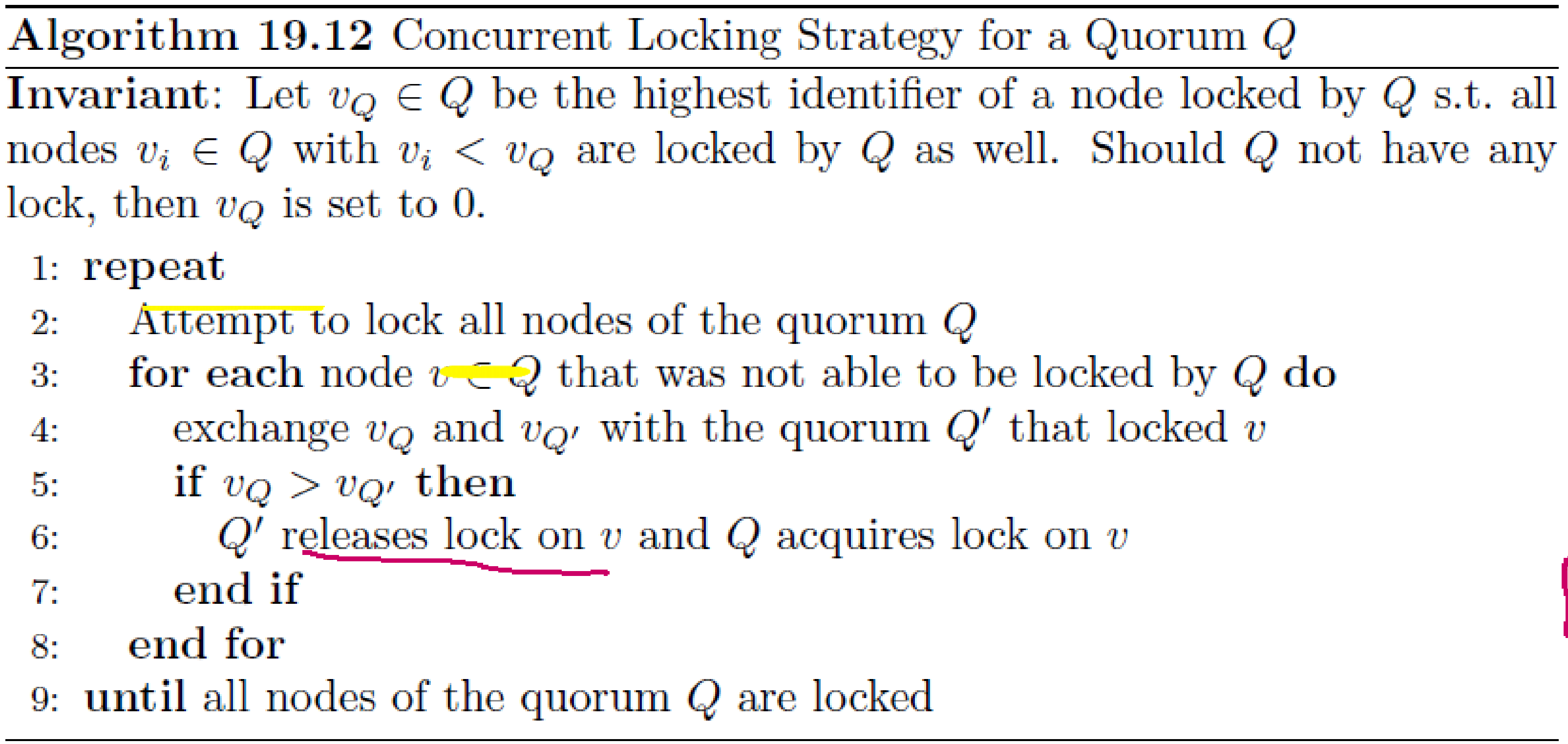
$x := x \cdot 2,\ x := x+1$

$0 \to 0 \to \underline{1}$





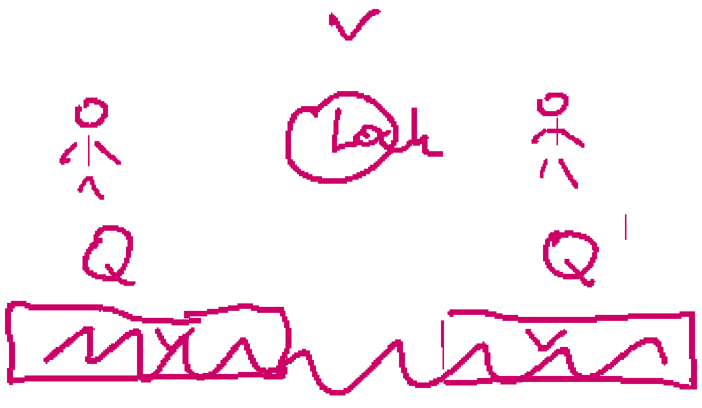
Different clients - locking a quorum



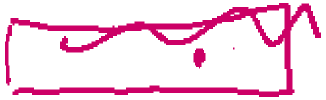
Algorithm 19.12 Concurrent Locking Strategy for a Quorum Q

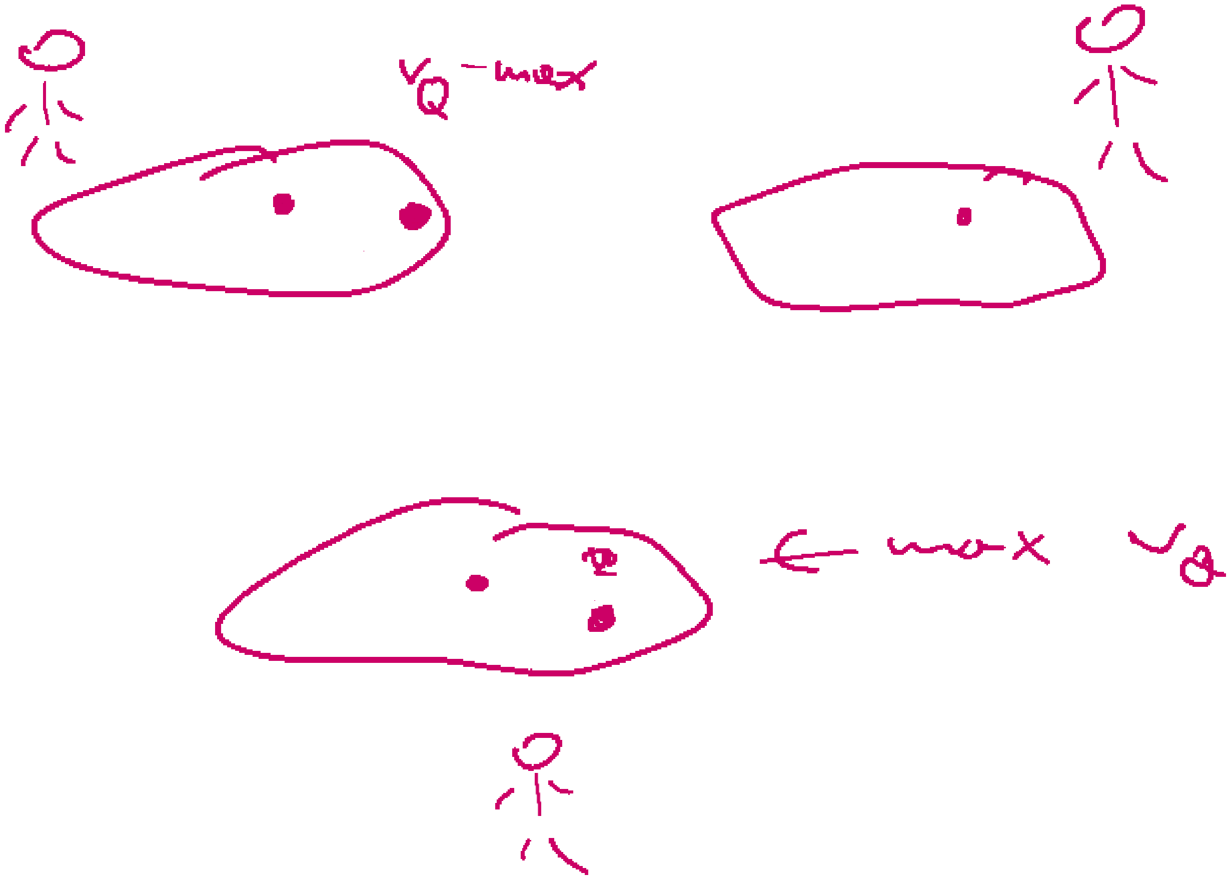
Invariant: Let $v_Q \in Q$ be the highest identifier of a node locked by Q s.t. all nodes $v_i \in Q$ with $v_i < v_Q$ are locked by Q as well. Should Q not have any lock, then v_Q is set to 0.

```
repeat
  Attempt to lock all nodes of the quorum $Q$
  for each node $v \in Q$ that was not able to be locked by $Q$ do
    exchange $v_Q$ and $v_{Q'}$ with the quorum $Q'$ that locked $v$
    if $v_Q > v_{Q'}$ then
      $Q'$ releases lock on $v$ and $Q$ acquires lock on $v$
    end if
  end for
until all nodes of the quorum $Q$ are locked
```



Lemma. At least one quorum will make progress





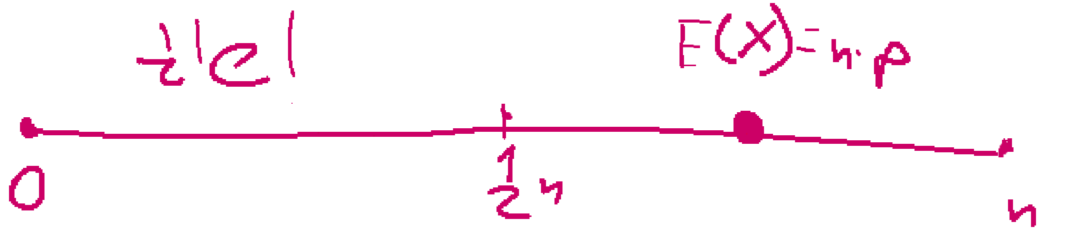
v_Q - max
← max v_Q

Fault tolerance

Assumption: some number of nodes may fail. (e.g.up to f nodes)

What is the probability that there is a quorum without failed nodes?

Majority



X = # serwerów żywych

- probability that a node is alive is *p*
- expected number of failed nodes is *n*p*
- system fails if at least *n/2* nodes fail
- By Chernoff bounds this is very unlikely if *p>0.5*

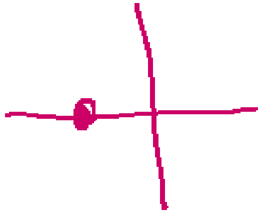
~~n=3, ..., 47~~

n – node

non-failed nodes: Ex $n \cdot p > \frac{n}{2}$

$$\Pr\left(\# \text{failed} \geq \frac{n}{2}\right) \xrightarrow[n\to\infty]{} 0$$

Grid



- It suffices to have one failed node in each row
- Let *d=SQRT(n)*

$$Pr[\text{at least one failure per row}] = \underbrace{(1-p^d)}_{}{}^d \geq 1 - dp^d \xrightarrow[n\to\infty]{} 1.$$

$$\neg\,\text{dobra} = 1 - Pr(\text{dobra}) = 1 - p^d$$

$$(1-p^d)^d \geq 1 - dp^d = 1 - \underbrace{\sqrt{n}\cdot p^{\sqrt{n}}}_{} \rightarrow 1$$

B-grid, example of failure tolerant architecture

$d \cdot r \cdot h = n$

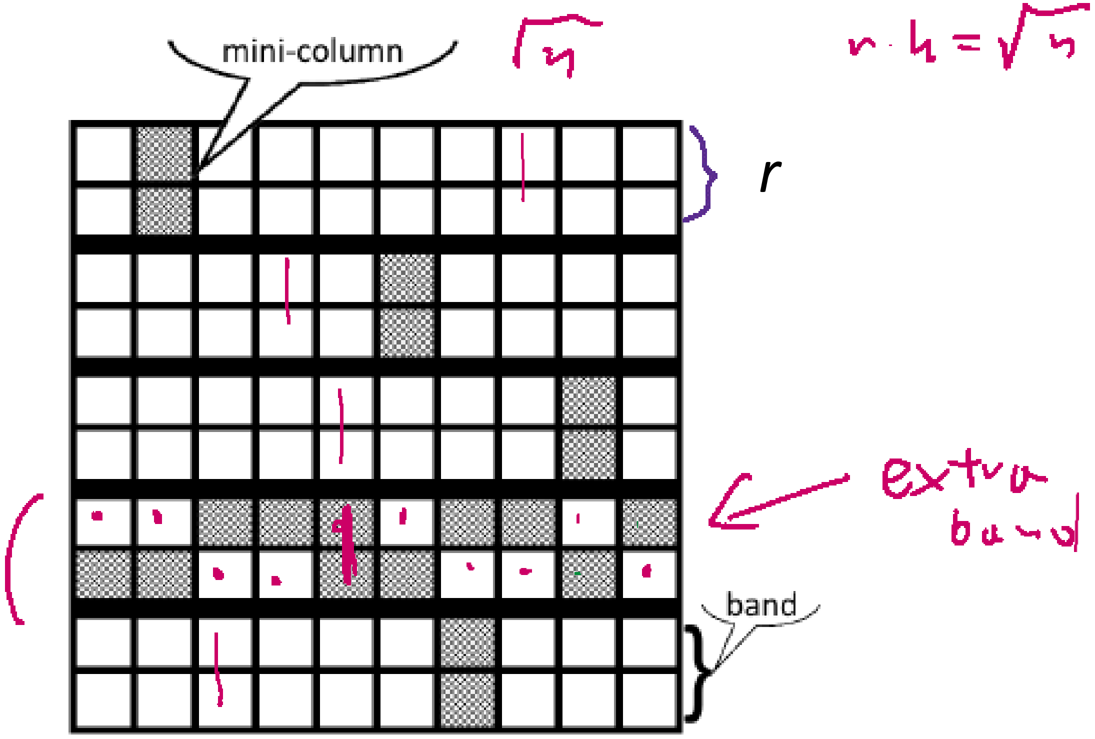
$r \cdot h = \sqrt{n}$

- *d* columns
- *r*h* rows ..
- .. divided into bands consisting of *r* rows

Choice of parameters:

d=SQRT(n)
r=ln(n)

Then failure probability goes to 0 with n



$$\left(1-\frac{1}{k}\right)^k = \frac{1}{e}$$