Cassiopeia
House of Computer Science

www.cs.aau.dk

# Quorum Systems

Material with complete proofs:

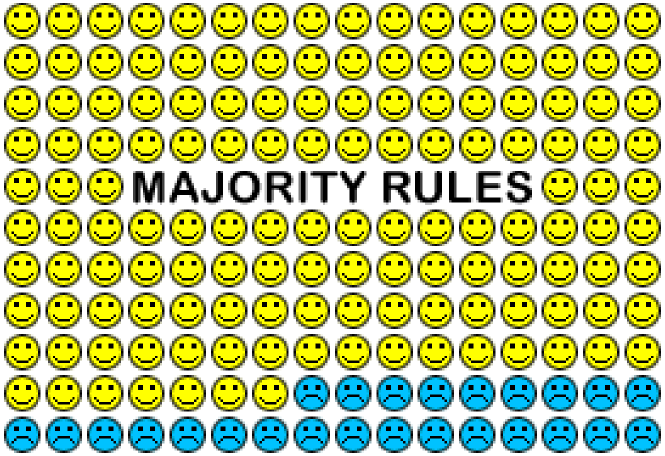
- Chapter 5: Quorum Systems (@https://disco.ethz.ch/courses/distsys/)

**Klaus-Tycho Foerster**

AALBORG UNIVERSITET

# Singleton (Primary Copy) and Majority

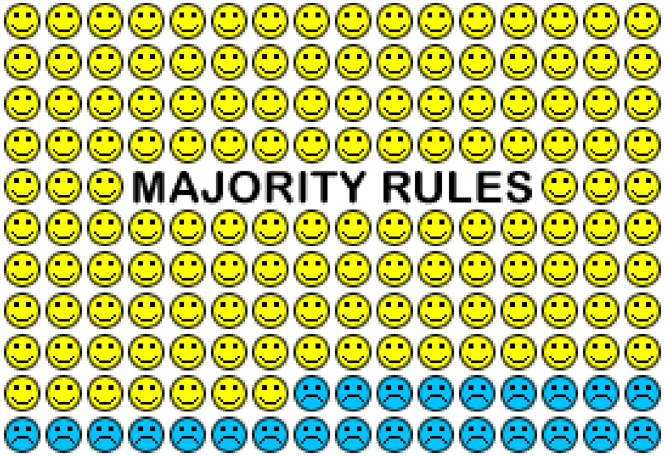


# Singleton (Primary Copy) and Majority

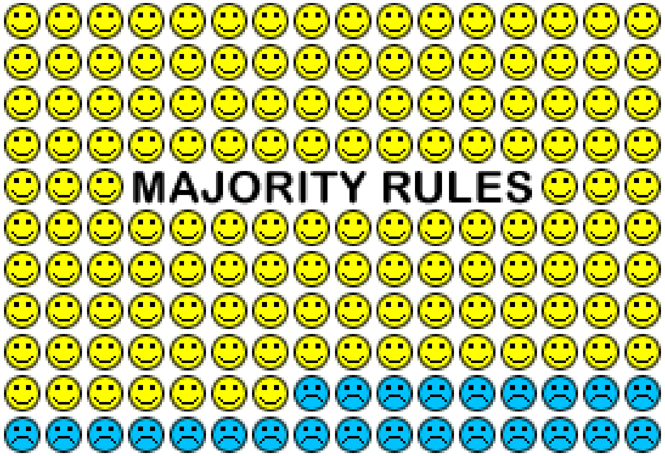


- A single server is great, but what about computational power?

# Singleton (Primary Copy) and Majority

- A single server is great, but what about computational power?
- Majority: still some server handles half the requests....

# Singleton (Primary Copy) and Majority

- A single server is great, but what about computational power?
- Majority: still some server handles half the requests....

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | $\approx 50\%$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ |

# Singleton (Primary Copy) and Majority



- A single server is great, but what about computational power?
- Majority: still some server handles half the requests....
- What can we do?

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ |

# Singleton (Primary Copy) and Majority

- A single server is great, but what about computational power?
- Majority: still some server handles half the requests….
- What can we do?
- Let‘s take a step back and be more formal ☺

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ |

# Quorum Introduction



Definition:

- Let $V = \{v_1, \ldots, v_n\}$ be a set of $n$ nodes
- A **quorum** $Q \subseteq V$ is a subset of these nodes.
- A **quorum system** $S \subset 2^V$ is a set of quorums such that every two quorums intersect
  - **Minimal** if no quorum is a proper subset of another

# Quorum Introduction



Definition:

- Let $V = \{v_1, \dots, v_n\}$ be a set of $n$ nodes
- A **quorum** $Q \subseteq V$ is a subset of these nodes.
- A **quorum system** $S \subset 2^V$ is a set of quorums such that every two quorums intersect
  - **Minimal** if no quorum is a proper subset of another

We will assume this

# Quorum Introduction



Definition:

- Let $V = \{v_1, \dots, v_n\}$ be a set of $n$ nodes
- A **quorum** $Q \subseteq V$ is a subset of these nodes.
- A **quorum system** $S \subset 2^V$ is a set of quorums such that every two quorums intersect

Why?

  - **Minimal** if no quorum is a proper subset of another

# Quorum Introduction



<u>Definition:</u>

- Let $V = \{v_1, \dots, v_n\}$ be a set of $n$ nodes
- A **quorum** $Q \subseteq V$ is a subset of these nodes.
- A **quorum system** $S \subset 2^V$ is a set of qu
  two quorums intersect
  - **Minimal** if no quorum is a proper su

Why?

**At least one node is up-to-date**

# Quorum Introduction



Definition:

- Let $V = \{v_1, \dots, v_n\}$ be a set of $n$ nodes
- A **quorum** $Q \subseteq V$ is a subset of these nodes.
- A **quorum system** $S \subset 2^V$ is a set of quorums such that every two quorums intersect
  - **Minimal** if no quorum is a proper subset of another

High-level functionality:

1. Client selects a free quorum
2. Locks all nodes of the quorum
3. Client releases all locks

# Load and Work

- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$

# Load and Work

- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$

- The **load** of $Z$ on a **node** $v_i$ is $L_Z(v_i) = \sum_{Q \in S; v_i \in Q} P_Z(Q)$

# Load and Work



- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$
- The **load** of $Z$ on a **node** $v_i$ is $L_Z(v_i) = \sum_{Q \in S; v_i \in Q} P_Z(Q)$
- The **load** of an **access strategy** $Z$ on a quorum system $S$ is the maximal load of $Z$ on any node in $S$: $L_Z(S) = max_{v_i \in S} L_Z(v_i)$

# Load and Work



- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$

- The **load** of $Z$ on a **node** $v_i$ is $L_Z(v_i) = \sum_{Q \in S; v_i \in Q} P_Z(Q)$

- The **load** of an **access strategy** $Z$ on a quorum system $S$ is the maximal load of $Z$ on any node in $S$: $L_Z(S) = max_{v_i \in S} L_Z(v_i)$

- The **load** of a **quorum system** $S$ is $L(S) = min_Z L_Z(S)$

# Load and Work

- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$
- The **load** of $Z$ on a **node** $v_i$ is $L_Z(v_i) = \sum_{Q \in S; v_i \in Q} P_Z(Q)$
- The **load** of an **access strategy** $Z$ on a quorum system $S$ is the maximal load of $Z$ on any node in $S$: $L_Z(S) = max_{v_i \in S} L_Z(v_i)$
- The **load** of a **quorum system** $S$ is $L(S) = min_Z L_Z(S)$

| | Singleton | Majority |
|---|---|---|
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

# Load and Work



- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$

- The **work** of a **quorum** $Q \in S$ is #nodes in $Q$, $W(Q) = |Q|$

# Load and Work



- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$

- The **work** of a **quorum** $Q \in S$ is #nodes in $Q$, $W(Q) = |Q|$

- The **work** of an **access strategy** $Z$ on a quorum system $S$ is the expected #nodes accessed: $W_Z(S) = \sum_{Q \in S} P_Z(Q) * W(Q)$

# Load and Work

- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$

- The **work** of a **quorum** $Q \in S$ is #nodes in $Q$, $W(Q) = |Q|$

- The **work** of an **access strategy** $Z$ on a quorum system $S$ is the expected #nodes accessed: $W_Z(S) = \sum_{Q \in S} P_Z(Q) * W(Q)$

- The **work** of a **quorum system** $S$ is $\mathrm{W}(S) = min_Z W_Z(S)$

# Load and Work

- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$

- The **work** of a **quorum** $Q \in S$ is #nodes in $Q$, $W(Q) = |Q|$

- The **work** of an **access strategy** $Z$ on a quorum system $S$ is the expected #nodes accessed: $W_Z(S) = \sum_{Q \in S} P_Z(Q) * W(Q)$

- The **work** of a **quorum system** $S$ is $\mathrm{W}(S) = min_Z W_Z(S)$

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |

# Load and Work

- An **access strategy** $Z$ defines the probability $P_Z(Q)$ of accessing a quorum $Q \in S$ such that $\sum_{Q \in S} P_Z(Q) = 1$
- The **work** of a **quorum** $Q \in S$ is #nodes in $Q$, $W(Q) = |Q|$
- The **work** of an **access strategy** $Z$ on a quorum system $S$ is the expected #nodes accessed: $W_Z(S) = \sum_{Q \in S} P_Z(Q) * W(Q)$
- The **work** of a **quorum system** $S$ is $\mathrm{W}(S) = min_Z W_Z(S)$

**Access strategy for work and load has to be the same**

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

# Example



- Let the nodes be $V = \{v_1, v_2, v_3, v_4, v_5\}$
- Set quorum system $S = \{Q_1, Q_2, Q_3, Q_4\}$ & access strategy $Z$ as:
  - $Q_1 = \{v_1, v_2\}, P_Z(Q_1) = \frac{1}{2}$
  - $Q_2 = \{v_1, v_3, v_4\}, \ P_Z(Q_2) = \frac{1}{6}$
  - $Q_3 = \{v_2, v_3, v_5\}, P_Z(Q_3) = \frac{1}{6}$
  - $Q_4 = \{v_2, v_4, v_5\}, P_Z(Q_4) = \frac{1}{6}$

# Example

- Let the nodes be $V = \{v_1, v_2, v_3, v_4, v_5\}$
- Set quorum system $S = \{Q_1, Q_2, Q_3, Q_4\}$ & access strategy $Z$ as:
  - $Q_1 = \{v_1, v_2\}, P_Z(Q_1) = \frac{1}{2}$
  - $Q_2 = \{v_1, v_3, v_4\}, \; P_Z(Q_2) = \frac{1}{6}$
  - $Q_3 = \{v_2, v_3, v_5\}, P_Z(Q_3) = \frac{1}{6}$
  - $Q_4 = \{v_2, v_4, v_5\}, P_Z(Q_4) = \frac{1}{6}$

- **Node** with **highest load**: ?
- **Work** of access strategy $S$ ?

# Example



- Let the nodes be $V = \{v_1, v_2, v_3, v_4, v_5\}$
- Set quorum system $S = \{Q_1, Q_2, Q_3, Q_4\}$ & access strategy $Z$ as:
  - $Q_1 = \{v_1, v_2\}$, $P_Z(Q_1) = \frac{1}{2}$
  - $Q_2 = \{v_1, v_3, v_4\}$, $P_Z(Q_2) = \frac{1}{6}$
  - $Q_3 = \{v_2, v_3, v_5\}$, $P_Z(Q_3) = \frac{1}{6}$
  - $Q_4 = \{v_2, v_4, v_5\}$, $P_Z(Q_4) = \frac{1}{6}$

- **Node** with **highest load**: $v_2$ with $L_Z(v_2) = \frac{1}{2} + \frac{1}{6} + \frac{1}{6} = \frac{5}{6}$

- **Work** of access strategy $S$

?

# Example

- Let the nodes be $V = \{v_1, v_2, v_3, v_4, v_5\}$
- Set quorum system $S = \{Q_1, Q_2, Q_3, Q_4\}$ & access strategy $Z$ as:
  - $Q_1 = \{v_1, v_2\}$, $P_Z(Q_1) = \frac{1}{2}$
  - $Q_2 = \{v_1, v_3, v_4\}$, $P_Z(Q_2) = \frac{1}{6}$
  - $Q_3 = \{v_2, v_3, v_5\}$, $P_Z(Q_3) = \frac{1}{6}$
  - $Q_4 = \{v_2, v_4, v_5\}$, $P_Z(Q_4) = \frac{1}{6}$

- **Node** with **highest load**: $v_2$ with $L_Z(v_2) = \frac{1}{2} + \frac{1}{6} + \frac{1}{6} = \frac{5}{6}$

- **Work**: $W_Z(S) = \frac{1}{2} * 2 + \frac{1}{6} * 3 + \frac{1}{6} * 3 + \frac{1}{6} * 3 = \frac{15}{6} = 2.5$

# Example

- Let the nodes be $V = \{v_1, v_2, v_3, v_4, v_5\}$
- Set quorum system $S = \{Q_1, Q_2, Q_3, Q_4\}$ & access strategy $Z$ as:
  - $Q_1 = \{v_1, v_2\}, P_Z(Q_1) = \frac{1}{2}$
  - $Q_2 = \{v_1, v_3, v_4\}, \ P_Z(Q_2) = \frac{1}{6}$
  - $Q_3 = \{v_2, v_3, v_5\}, P_Z(Q_3) = \frac{1}{6}$
  - $Q_4 = \{v_2, v_4, v_5\}, P_Z(Q_4) = \frac{1}{6}$

**Home Exercise: Improve access strategy (and S?)**

- **Node** with **highest load**: $v_2$ with $L_Z(v_2) = \frac{1}{2} + \frac{1}{6} + \frac{1}{6} = \frac{5}{6}$
- **Work**: $W_Z(S) = \frac{1}{2} * 2 + \frac{1}{6} * 3 + \frac{1}{6} * 3 + \frac{1}{6} * 3 = \frac{15}{6} = 2.5$

# How good can the load get?

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

# How good can the load get?

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

- Maybe surprisingly: Can't be better than $1/\sqrt{n}$

# How good can the load get?



| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

- Maybe surprisingly: Can't be better than $\mathbf{1}/\sqrt{\boldsymbol{n}}$

**Proof Sketch:**

1. Pick a quorum $Q$ of minimum size $\boldsymbol{q}$ in the quorum system $S$

# How good can the load get?

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

- Maybe surprisingly: Can't be better than $\mathbf{1}/\sqrt{\boldsymbol{n}}$

**Proof Sketch:**

1. Pick a quorum $Q$ of minimum size $\boldsymbol{q}$ in the quorum system $S$
2. When *any* quorum is accessed by $Z$, one node from $Q$ as well
   - Lower bound of $\boldsymbol{L_Z(v_i)} \geq \mathbf{1}/\boldsymbol{q}$ for some $v_i \in Q$

# How good can the load get?

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

- Maybe surprisingly: Can't be better than $\mathbf{1}/\sqrt{\boldsymbol{n}}$

**Proof Sketch:**

1. Pick a quorum $Q$ of minimum size $\boldsymbol{q}$ in the quorum system $S$
2. When *any* quorum is accessed by $Z$, one node from $Q$ as well
   - Lower bound of $\boldsymbol{L_Z(v_i)} \geq \mathbf{1}/\boldsymbol{q}$ for some $v_i \in Q$
3. $Q$ minimum → each access of $Z$ uses $q$ nodes, $\boldsymbol{W(S)} \geq \boldsymbol{q}$
   - Access $\geq q$ nodes? Load of most accessed node $\geq q/n$

# How good can the load get?

| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

- Maybe surprisingly: Can't be better than $\mathbf{1}/\sqrt{\boldsymbol{n}}$

**Proof Sketch:**

1. Pick a quorum $Q$ of minimum size $\boldsymbol{q}$ in the quorum system $S$
2. When *any* quorum is accessed by $Z$, one node from $Q$ as well
   - Lower bound of $\boldsymbol{L_Z(v_i)} \geq \mathbf{1}/\boldsymbol{q}$ for some $v_i \in Q$
3. $Q$ minimum → each access of $Z$ uses $q$ nodes, $\boldsymbol{W(S)} \geq \boldsymbol{q}$
   - Access $\geq q$ nodes? Load of most accessed node $\geq q/n$
4. Combine: $L_Z(S) \geq \max({}^1/_q, {}^q/_n)$ implies $L_Z(S) \geq {}^1/_{\sqrt{n}}$

# How good can the load get?



| | Singleton | Majority |
|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ |
| What's the load of the busiest server? (**Load**) | 100% | ≈ 50% |

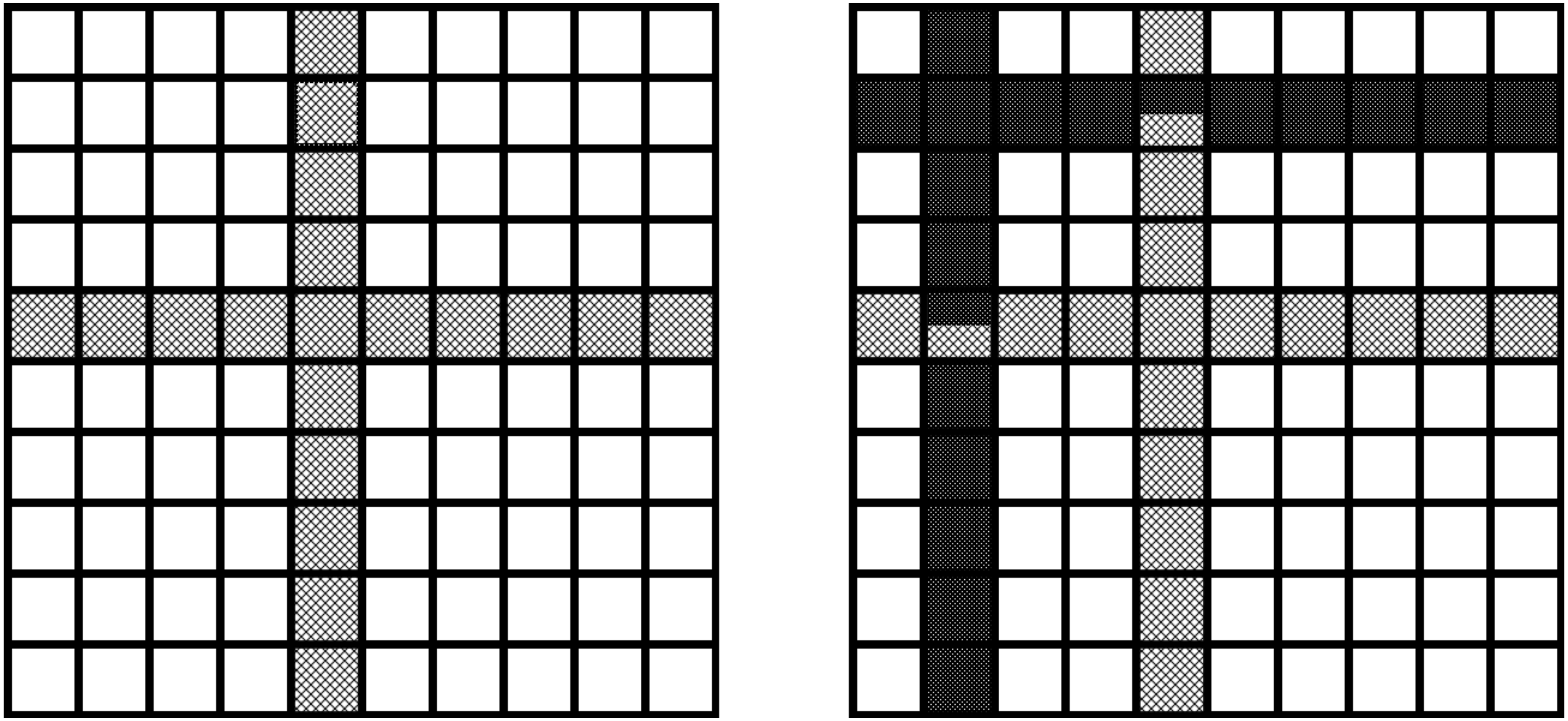
- Maybe surprisingly: Can't be better than $\mathbf{1/\sqrt{n}}$

**Proof Sketch:**

**Can we reach this bound?**

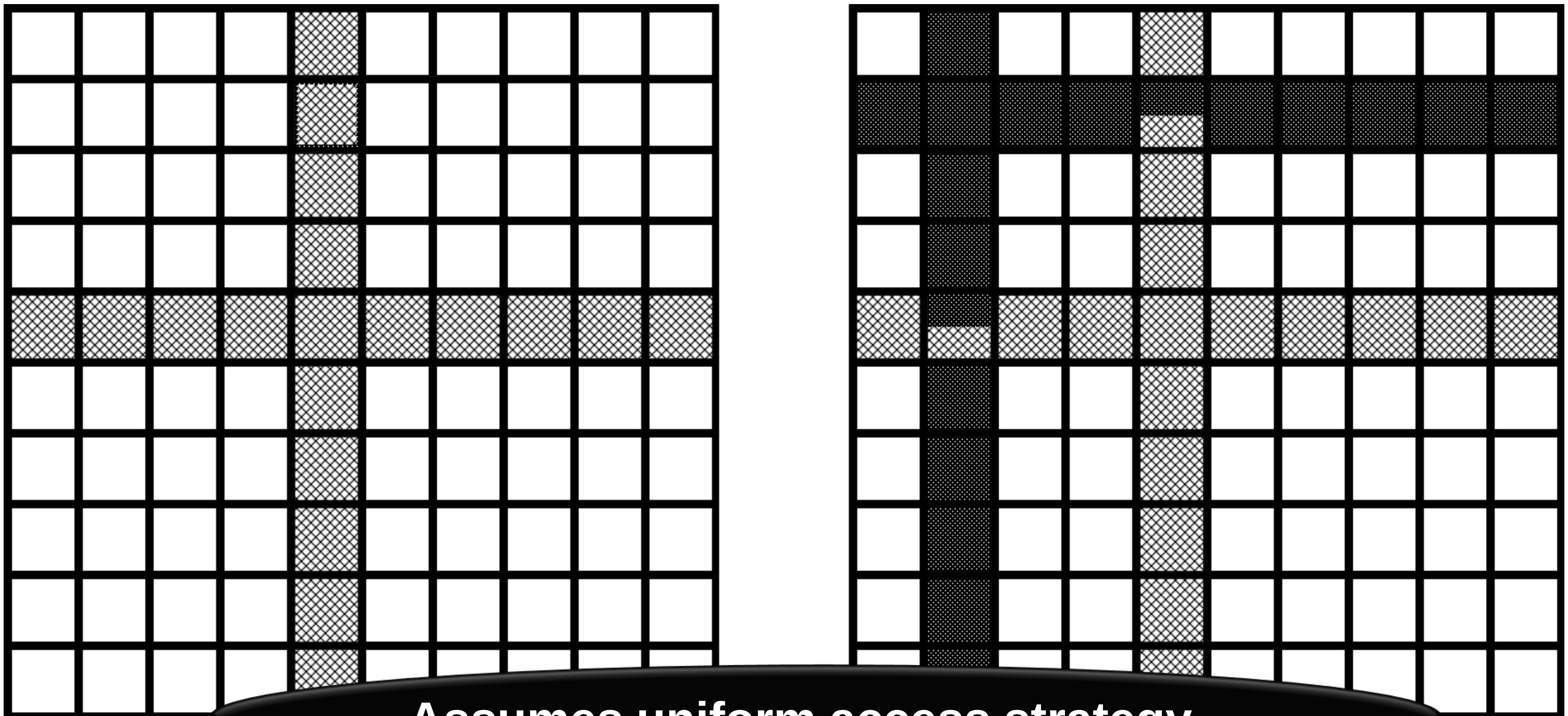
1. Pick a quorum
2. When *any* quorum is accessed by $Z$, one node from $Q$ as well
   - Lower bound of $\boldsymbol{L_Z(v_i) \geq 1/q}$ for some $v_i \in Q$
3. $Q$ minimum → each access of $Z$ uses $q$ nodes, $\boldsymbol{W(S) \geq q}$
   - Access $\geq q$ nodes? Load of most accessed node $\geq q/n$
4. Combine: $L_Z(S) \geq \max(1/q, q/n)$ implies $L_Z(S) \geq 1/\sqrt{n}$

# Grid Quorum Systems



# Grid Quorum Systems

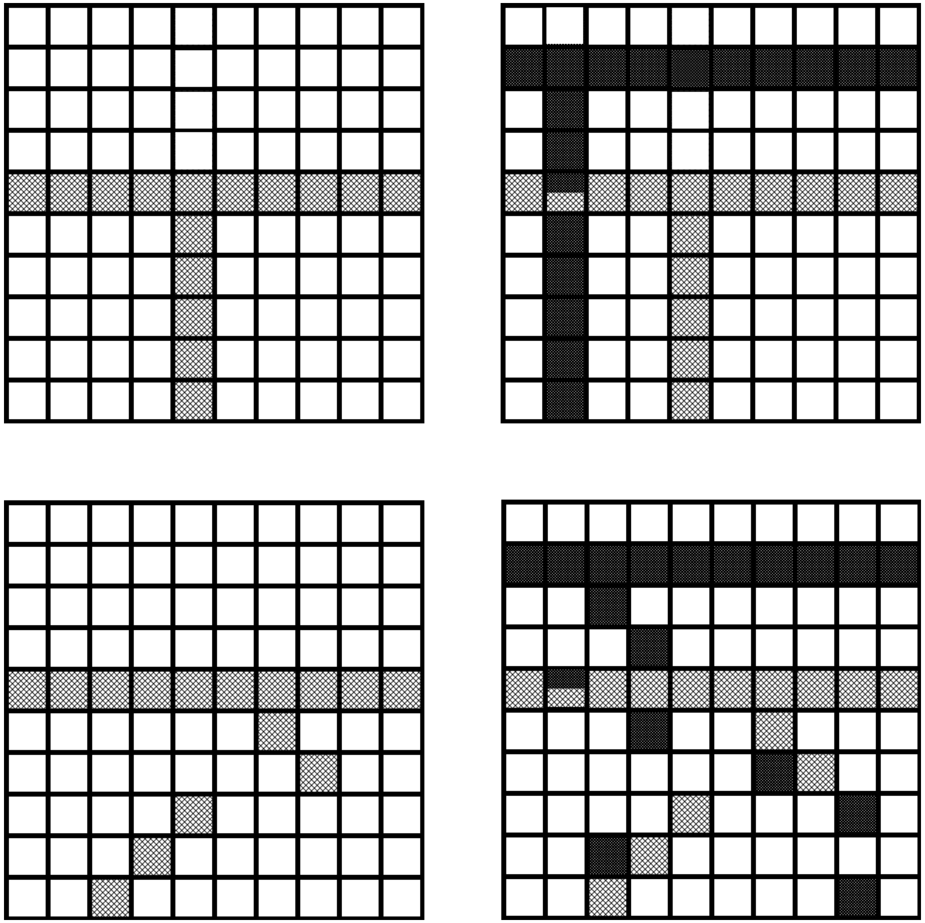




**Assumes uniform access strategy**

Work: $2\sqrt{n} - 1$, Load: $\frac{2\sqrt{n}-1}{n}$

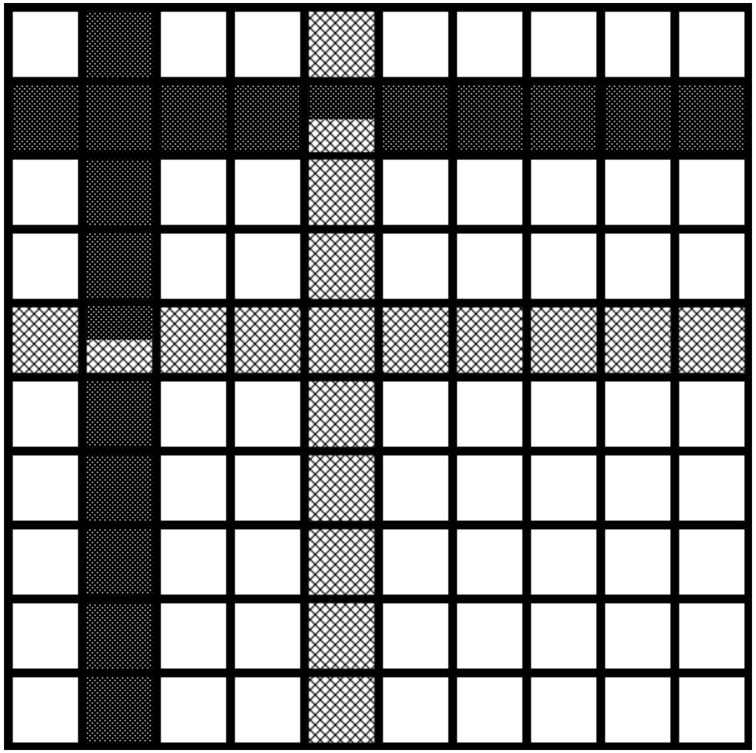
# Grid Quorum Systems



# Who gets access?

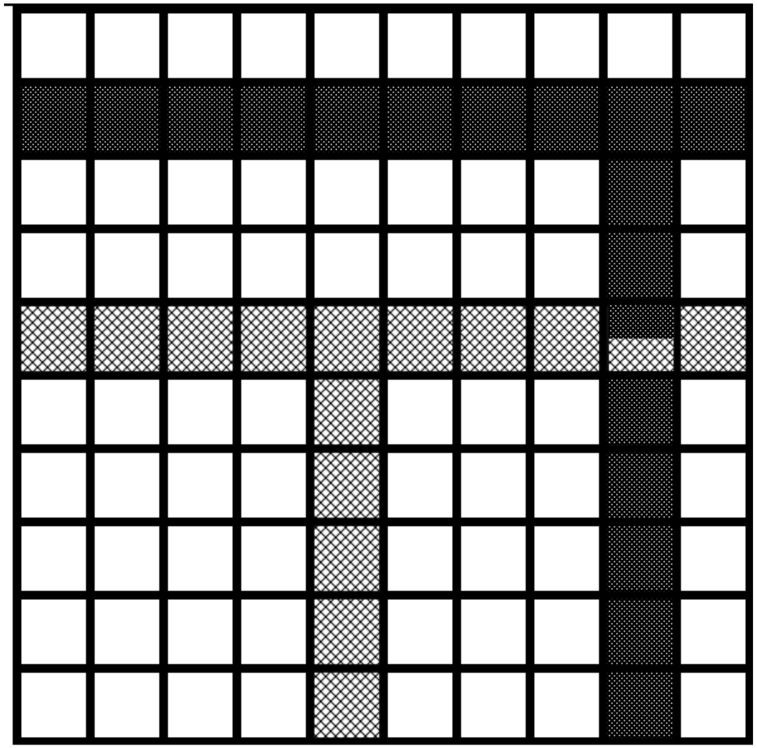


- Who can lock his quorum?



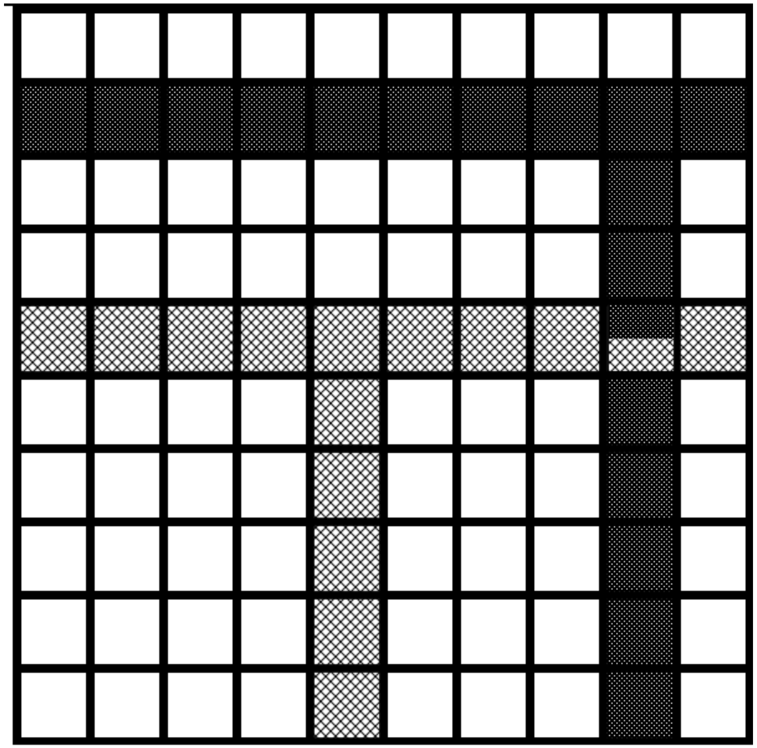
# Who gets access?

- Who can lock his quorum?
- Overlap only in 1 node?



# Who gets access?

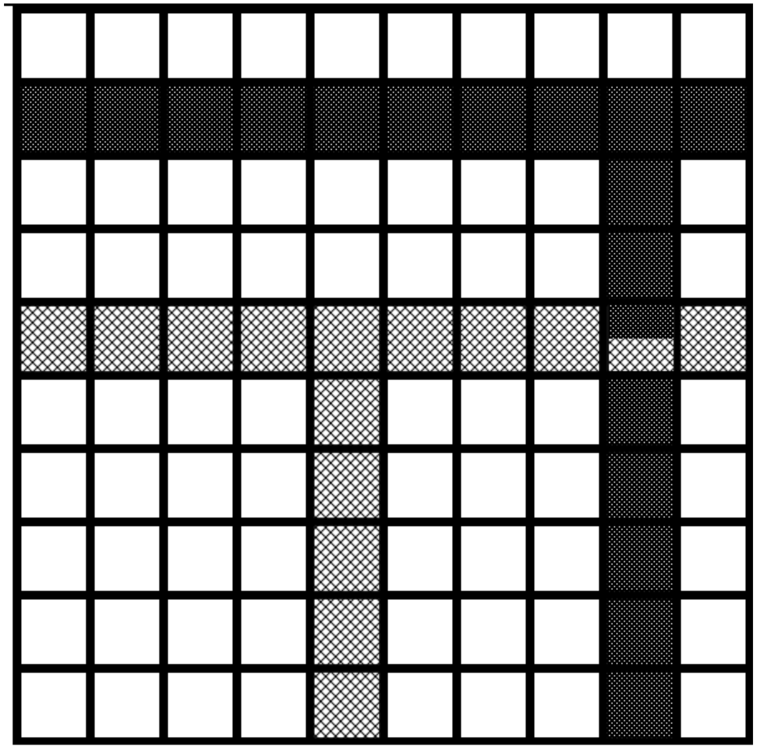
- Who can lock his quorum?
- Overlap only in 1 node?
  - Same problem for 3 quorums



# Who gets access?



- Who can lock his quorum?
- Overlap only in 1 node?
  - Same problem for 3 quorums



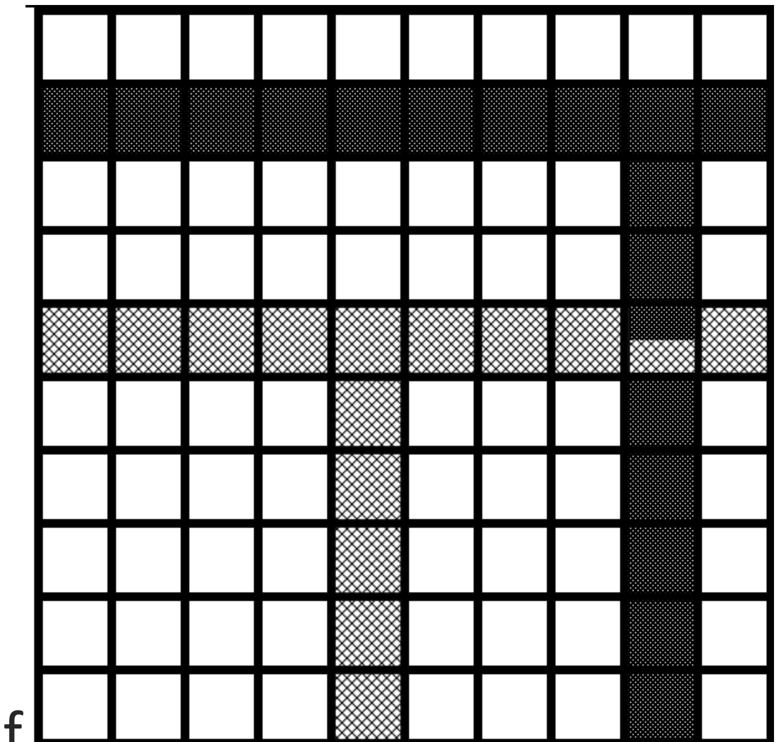
**Sequential locking idea**

- Fix for each node an unique id
- Lock nodes sorted by ids
  - Fail? Start over
- Idea: Someone gets highest id.

# Who gets access?



- Who can lock his quorum?
- Overlap only in 1 node?
  - Same problem for 3 quorums



**Sequential locking idea**

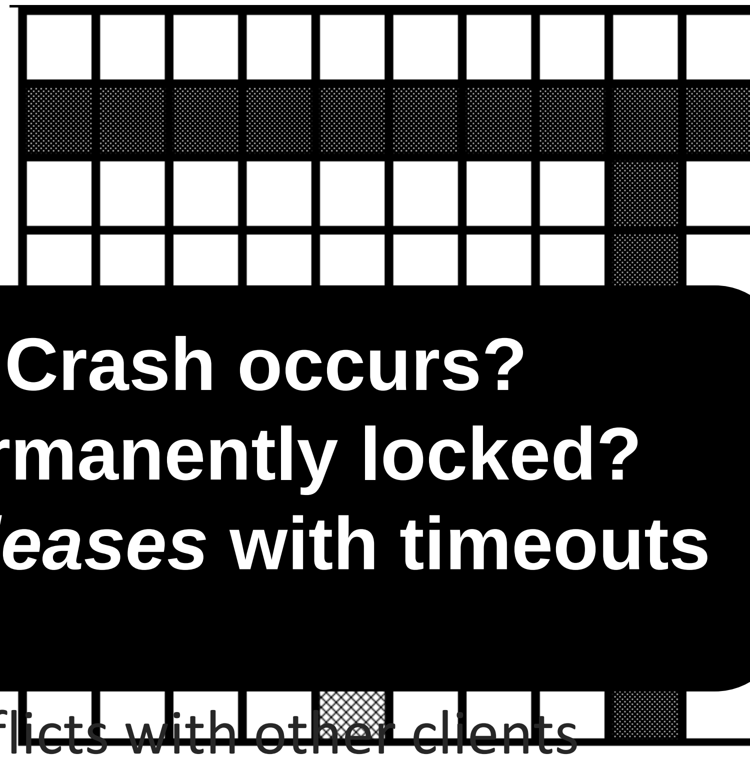
- Fix for each node an unique id
- Lock nodes sorted by ids
  - Fail? Start over
- Idea: Someone gets highest id.

**Add concurrency:**

- Lock as many as you can, break conf
- Idea: At least one succeeds due to highest id.

# Who gets access?

- Who can lock his quorum?
- Overlap only in 1 node?
  - Same problem for 3 quorums



**Sequential locking idea**

- Fix for each node an uniqu
- Lock nodes sorted by ids
  - Fail? Start over
- Idea: Someone gets highes

**Crash occurs?**
**Permanently locked?**
**Use *leases* with timeouts**

**Add concurrency:**

- Lock as many as you can, break conflicts with other clients
- Idea: At least one succeeds due to highest id.

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ | $\Theta(\sqrt{n})$ |

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ | $\Theta(\sqrt{n})$ |

- However, resilience is sort of worst case…

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ | $\Theta(\sqrt{n})$ |

- However, resilience is sort of worst case...
- Assume that every node works with a fixed probability $p > 1/2$

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ | $\Theta(\sqrt{n})$ |

- However, resilience is sort of worst case…
- Assume that every node works with a fixed probability $p > 1/2$
- The **failure probability** $F_P(S)$ of a quorum system is the probability that at least one node of every quorum fails

AALBORG UNIVERSITET

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ | $\Theta(\sqrt{n})$ |

- However, resilience is sort of worst case…
- Assume that every node works with a fixed probability $p > 1/2$
- The **failure probability** $F_P(S)$ of a quorum system is the probability that at least one node of every quorum fails
- Asymptotic failure probability: $F_P(S)$ for $n \to \infty$

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ | $\Theta(\sqrt{n})$ |

- Lets look at the asymptotic failure probabilities ☺

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | 1 | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< n/2$ | $\Theta(\sqrt{n})$ |

- Lets look at the asymptotic failure probabilities ☺
- Singleton? $1 - p$

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | $1$ | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | $1$ | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | $0$ | $< n/2$ | $\Theta(\sqrt{n})$ |

- Lets look at the asymptotic failure probabilities ☺
- Singleton? $1 - p$
- Majority? 0
  - Half of the nodes must fail… ☺

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | $1$ | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | $1$ | $> 1/2$ | $\Theta(1/\sqrt{n})$ |
| How many server failures can be tolerated? (**Resilience**) | $0$ | $< n/2$ | $\Theta(\sqrt{n})$ |

- Lets look at the asymptotic failure probabilities ☺
- Singleton? $1 - p$
- Majority? 0
  - Half of the nodes must fail… ☺
- Grid? 1
  - Single failure per row is enough ☹

# Fault Tolerance



| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | **1** | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< \boldsymbol{n/2}$ | $\Theta(\sqrt{n})$ |
| Asymptotic **Failure Probability** | $1 - p$ | **0** | 1 |

- Lets look at the asymptotic failure probabilities ☺
- Singleton? $1 - p$
- Majority? 0
  - Half of the nodes must fail… ☺
- Grid? 1
  - Single failure per row is enough ☹

# Fault Tolerance

| | Single | Majority | Grid |
|---|---|---|---|
| How many servers need to be contacted? (**Work**) | **1** | $> n/2$ | $\Theta(\sqrt{n})$ |
| What's the load of the busiest server? (**Load**) | 1 | $> 1/2$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ |
| How many server failures can be tolerated? (**Resilience**) | 0 | $< \boldsymbol{n/2}$ | $\Theta(\sqrt{n})$ |
| Asymptotic **Failure Probability** | $1 - p$ | **0** | 1 |

- Lets look at the asymptotic failure probabilities ☺
- Singleton? $1 - p$
- Majority? 0
  - Half of the nodes must fail… ☺
- Grid? 1
  - Single failure per row is enough ☹

**Can we get a system with good *Load* *Fault Tolerance* ?**

# The B-Grid



- Situation: Grid has optimal load, but bad failure resiliency
  - Single error per row suffices ☹

# The B-Grid

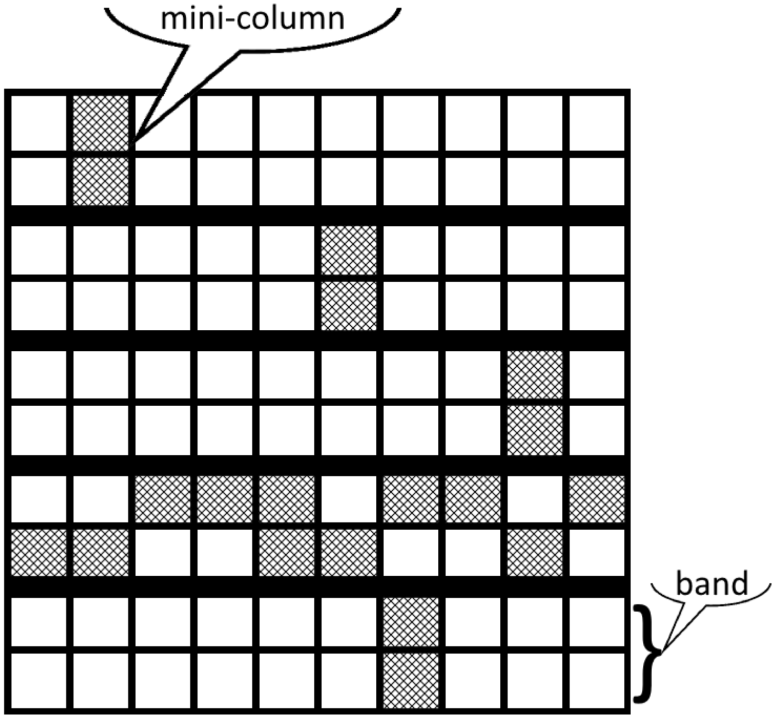


- Situation: Grid has optimal load, but bad failure resiliency
  - Single error per row suffices ☹
- Idea: Make rows thicker → a "band"
  - Whole band must fail

# The B-Grid

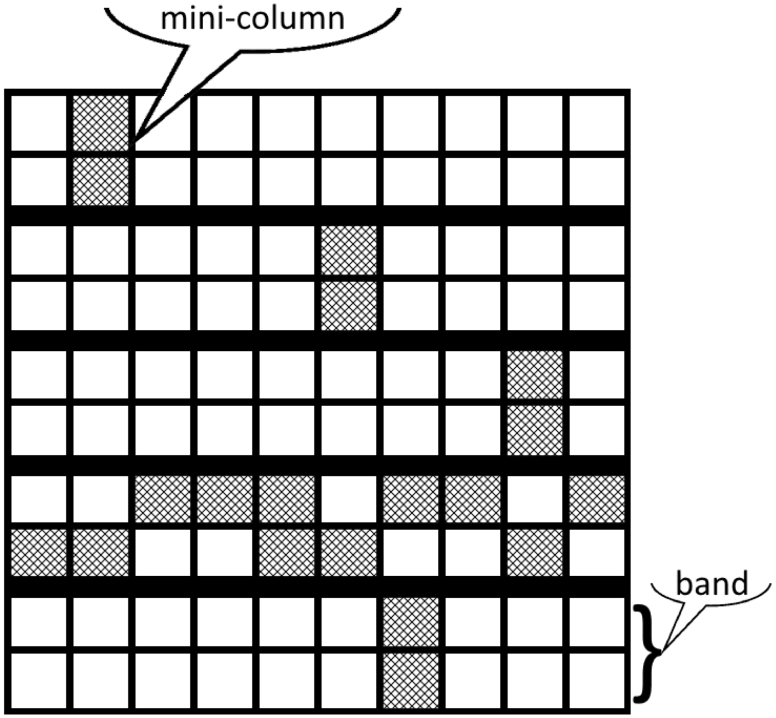
- Situation: Grid has optimal load, but bad failure resiliency
  - Single error per row suffices ☹

- Idea: Make rows thicker → a “band”
  - Whole band must fail



# The B-Grid

- Situation: Grid has optimal load, but bad failure resiliency
  - Single error per row suffices ☹
- Idea: Make rows thicker → a "band"
  - Whole band must fail
- Set band-thickness to $\log n$
  - Failure prob. → $0$



# Overview

| | Single | Majority | Grid | B-Grid |
|---|---|---|---|---|
| **Work** | **1** | $> n/2$ | $\Theta(\sqrt{n})$ | $\Theta(\sqrt{n})$ |
| **Load** | 1 | $> 1/2$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ |
| **Resilience** | 0 | $\boldsymbol{< n/2}$ | $\Theta(\sqrt{n})$ | $\Theta(\sqrt{n})$ |
| **Failure Probability** | $1 - p$ | →**0** | →1 | →**0** |

# Overview

| | Single | Majority | Grid | B-Grid |
|---|---|---|---|---|
| **Work** | **1** | $> n/2$ | $\Theta(\sqrt{n})$ | $\Theta(\sqrt{n})$ |
| **Load** | 1 | $> 1/2$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ |
| **Resilience** | 0 | $< \boldsymbol{n/2}$ | $\Theta(\sqrt{n})$ | $\Theta(\sqrt{n})$ |
| **Failure Probability** | $1 - p$ | →**0** | →1 | →**0** |

**What about malicious failures?**

# Disseminating Quorum Systems

1st Model: Nodes might provide false data, but can be detected (data is self-verifying)

# Disseminating Quorum Systems

1st Model: Nodes might provide false data, but can be detected (data is self-verifying)

- A quorum system $S$ is $f$-**disseminating**, if
  1. Intersection of 2 quorums always has $\geq f+1$ nodes
  2. for any $f$ byzantine nodes, one quorum is byzantine-free

# Disseminating Quorum Systems



1st Model: Nodes might provide false data, but can be detected (data is self-verifying)

- A quorum system $S$ is $f$-**disseminating**, if
  1. Intersection of 2 quorums always has $\geq f + 1$ nodes
  2. for any $f$ byzantine nodes, one quorum is byzantine-free

**Why?**

1. At least 1 node is not byzantine in every quorum intersection
2. Byzantine nodes crashing doesn't stop the quorum system

# Disseminating Quorum Systems

1st Model: Nodes might provide false data, but can be detected (data is self-verifying)

- A quorum system $S$ is $f$-**disseminating**, if
  1. Intersection of 2 quorums always has $\geq f+1$ nodes
  2. for any $f$ byzantine nodes, one quorum is byzantine-free

**Why?**

1. At least 1 node is not byzantine in every quorum intersection
2. Byzantine nodes crashing doesn't stop the quorum system

Lower load bound:

- $L(S) \geq \sqrt{(f+1)/n}$

# Disseminating Quorum Systems



1st Model: Nodes might provide false data, but can be detected (data is self-verifying)

- A quorum system $S$ is $f$-**disseminating**, if
  1. Intersection of 2 quorums always has $\geq f + 1$ nodes
  2. for any $f$ byzantine nodes, one quorum is byzantine-free

**Why?**

1. At least 1 node is not byzantine in every quorum intersection
2. Byzantine nodes crashing doesn't stop the quorum system

Lower load bound:

- $L(S) \geq \sqrt{(f+1)/n}$

**Proof: As before, but access $f + 1$ nodes each time**

# Disseminating Quorum Systems

1st Model: Nodes might provide false data, but can be detected (data is self-verifying)

**What if not?**

- A quorum system $S$ is $f$-**disseminating**, if
  1. Intersection of 2 quorums always has $\geq f + 1$ nodes
  2. for any $f$ byzantine nodes, one quorum is byzantine-free

**Why?**

1. At least 1 node is not byzantine in every quorum intersection
2. Byzantine nodes crashing doesn't stop the quorum system

Lower load bound:

- $L(S) \geq \sqrt{(f+1)/n}$

**Proof: As before, but access $f + 1$ nodes each time**

# Masking Quorum Systems

2nd Model: Solve data-conflicts by voting

- A quorum system $S$ is $f$-**masking**, if
  1. Intersection of 2 quorums always has $\geq \mathbf{2}f + 1$ nodes
  2. for any $f$ byzantine nodes, one quorum is byzantine-free

# Masking Quorum Systems



2nd Model: Solve data-conflicts by voting

- A quorum system $S$ is $f$-**masking**, if
  1. Intersection of 2 quorums always has $\geq \mathbf{2}f + 1$ nodes
  2. for any $f$ byzantine nodes, one quorum is byzantine-free

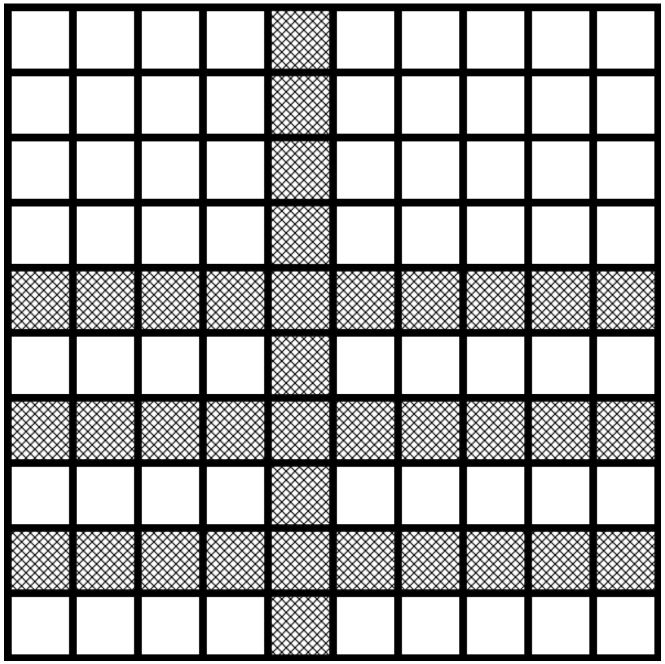
- $f$-masking is $\mathbf{2}f$-disseminating

Lower load bound:

- $L(S) \geq \sqrt{(\mathbf{2}f + 1)/n}$

# $f$-masking Grid

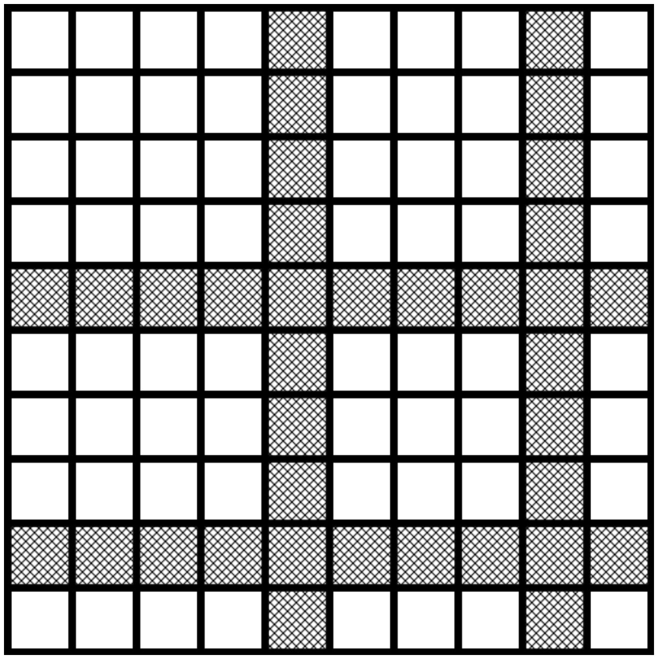
- Idea: pick $f+1$ rows in the Grid, overlap of at least $2f+1$



Load: $\Theta\left(f/\sqrt{n}\right)$

# M-Grid Quorum System

- Idea: $\sqrt{f+1}$ rows and columns in the Grid (overlap $\geq 2f+1$)



Load: $\Theta\left(\sqrt{f/n}\right)$

# Last one… (just intuition)



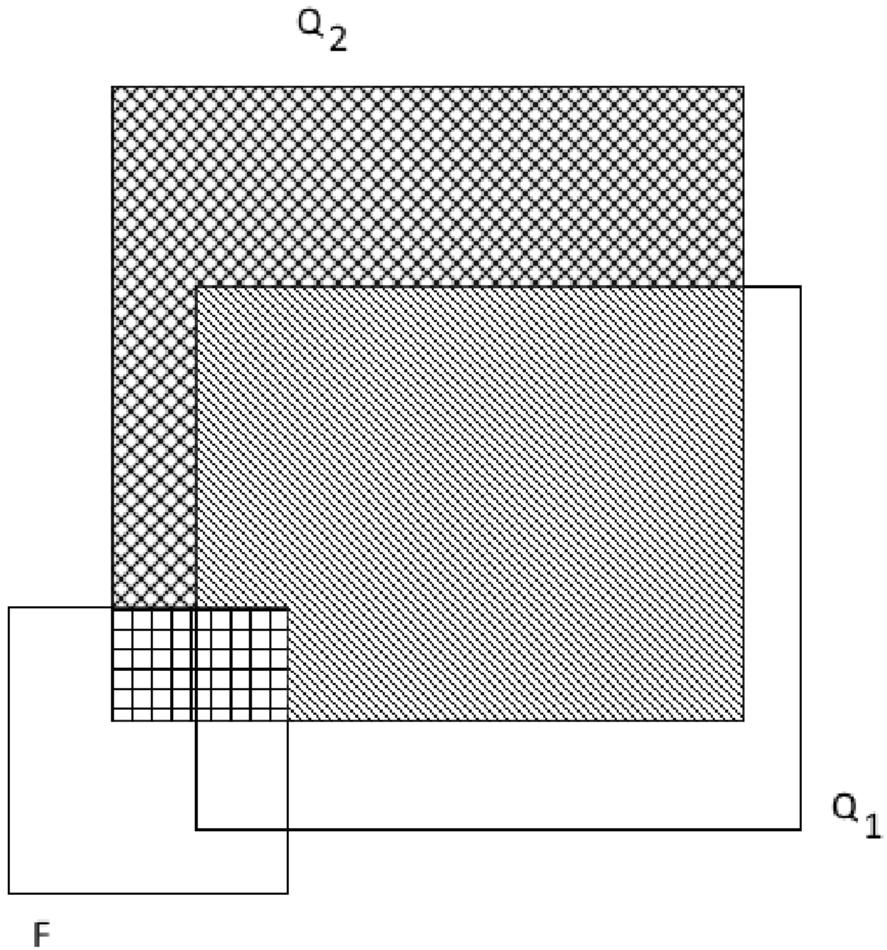
Load differences with byzantines don't look so bad…

- What happens if we access a quorum that is only up-to-date:
    - For the intersection with an up-to-date quorum? (change/write didn't propagate yet etc.)

- More up-to-date than out-of-date? (voting again..)

- Solved by so-called $f$-**opaque** systems

# $f$-Opaque systems




**Idea: failures don‘t destroy majority overlap**

# $f$-Opaque systems

- Majority can be extended to $f$-opaque:
  - Size of each quorum: $\geq \lceil (2n + 2f)/3 \rceil$
  - Load: $\frac{\lceil (2n+2f)/3 \rceil}{n} \approx {}^{2}/_{3} + {}^{2f}/_{3n} \geq 2/3$
  - However: Load of any $f$-opaque system is at least $1/2$
- Restrictions to number of byzantine failures:
  - $f$-opaque: $n > 5f$
  - $f$-masking: $n > 4f$
  - $f$-disseminating: $n > 3f$

# Looking back

| | Single | Majority | Grid | B-Grid |
|---|---|---|---|---|
| **Work** | **1** | $> n/2$ | $\Theta(\sqrt{n})$ | $\Theta(\sqrt{n})$ |
| **Load** | 1 | $> 1/2$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ | $\boldsymbol{\Theta(1/\sqrt{n})}$ |
| **Resilience** | 0 | $< \boldsymbol{n/2}$ | $\Theta(\sqrt{n})$ | $\Theta(\sqrt{n})$ |
| **Failure Probability** | $1-p$ | →**0** | →1 | →**0** |

# Quorum Systems

Material with complete proofs:

- Chapter 5: Quorum Systems (@https://disco.ethz.ch/courses/distsys/)

Klaus-Tycho Foerster