Chapter 15
Quorum Systems

What happens if a single server is no longer powerful enough to service all your customers? The obvious choice is to add more servers and to use the majority approach (e.g., Paxos, Chapter 7) to guarantee consistency. However, even if you buy one million servers, a client still has to access more than half of them per request! While you gain fault-tolerance, your efficiency can at most be doubled. Do we have to give up on consistency?

Let us take a step back: We used majorities because majority sets always overlap. But are majority sets the only sets that guarantee overlap? In this chapter we study the theory behind overlapping sets, known as quorum systems.

Definition 15.1 (quorum, quorum system). Let \( V = \{v_1, \ldots, v_n\} \) be a set of nodes. A quorum \( Q \subseteq V \) is a subset of these nodes. A quorum system \( S \subseteq 2^V \) is a set of quorums s.t. every two quorums intersect, i.e., \( Q_1 \cap Q_2 \neq \emptyset \) for all \( Q_1, Q_2 \in S \).

Remarks:
- When a quorum system is being used, a client selects a quorum, acquires a lock (or ticket) on all nodes of the quorum, and when done releases all locks again. The idea is that no matter which quorum is chosen, its nodes will intersect with the nodes of every other quorum.
- What can happen if two quorums try to lock their nodes at the same time?
- A quorum system \( S \) is called minimal if \( \forall Q_1, Q_2 \in S : Q_1 \not\subseteq Q_2 \).
- The simplest quorum system imaginable consists of just one quorum, which in turn just consists of one server. It is known as Singleton.
- In the Majority quorum system, every quorum has \( \lceil \frac{n}{2} \rceil + 1 \) nodes.
- Can you think of other simple quorum systems?

15.1 Load and Work

Definition 15.2 (access strategy). An access strategy \( Z \) defines the probability \( P_Z(Q) \) of accessing a quorum \( Q \in S \) s.t. \( \sum_{Q \in S} P_Z(Q) = 1 \).

Definition 15.3 (load).
- The load of access strategy \( Z \) on a node \( v \) is \( L_Z(v) = \sum_{Q \in S, v \in Q} P_Z(Q) \).
- The load induced by access strategy \( Z \) on a quorum system \( S \) is the maximal load induced by \( Z \) on any node in \( S \), i.e., \( L_Z(S) = \max_{v \in S} L_Z(v) \).
- The load of a quorum system \( S \) is \( L(S) = \min_Z L_Z(S) \).

Definition 15.4 (work).
- The work of a quorum \( Q \in S \) is the number of nodes in \( Q \), \( W(Q) = |Q| \).
- The work induced by access strategy \( Z \) on a quorum system \( S \) is the expected number of nodes accessed, i.e., \( W_Z(S) = \sum_{Q \in S} P_Z(Q) \cdot W(Q) \).
- The work of a quorum system \( S \) is \( W(S) = \min_Z W_Z(S) \).

Remarks:
- Note that you cannot choose different access strategies \( Z \) for work and load, you have to pick a single \( Z \) for both.
- We illustrate the above concepts with a small example. Let \( V = \{v_1, v_2, v_3, v_4, v_5\} \) and \( S = \{Q_1, Q_2, Q_3, Q_4\} \), with \( Q_1 = \{v_1, v_2\} \), \( Q_2 = \{v_3, v_4\} \), \( Q_3 = \{v_1, v_3, v_4\} \), \( Q_4 = \{v_2, v_3, v_5\} \). If we choose the access strategy \( Z \) s.t. \( P_Z(Q_1) = 1/2 \) and \( P_Z(Q_2) = P_Z(Q_3) = P_Z(Q_4) = 1/6 \), then the node with the highest load is \( v_2 \) with \( L_Z(v_2) = 1/2 + 1/6 + 1/6 = 5/6 \), i.e., \( L_Z(S) = 5/6 \). Regarding work, we have \( W_Z(S) = 1/2 \cdot 1/2 = 1/2 \) and \( L(Z(S)) = 1/2 \cdot 3 + 1/6 \cdot 3 + 1/6 \cdot 3 = 15/6 \).
- Can you come up with a better access strategy for \( S \)?
- If every quorum \( Q \) in a quorum system \( S \) has the same number of elements, \( S \) is called uniform.
- What is the minimum load a quorum system can have?

Table 15.5: First comparison of the Singleton and Majority quorum systems.

<table>
<thead>
<tr>
<th>Primary Copy vs. Majority</th>
<th>Singleton</th>
<th>Majority</th>
</tr>
</thead>
<tbody>
<tr>
<td>How many nodes need to be accessed? (Work)</td>
<td>1</td>
<td>&gt; n/2</td>
</tr>
<tr>
<td>What is the load of the busiest node? (Load)</td>
<td>1</td>
<td>&gt; 1/2</td>
</tr>
</tbody>
</table>

Note that the Singleton quorum system can be a good choice when the failure probability of every single node is > 1/2.
Theorem 15.6. Let $S$ be a quorum system. Then $L(S) \geq 1/\sqrt{n}$ holds.

Proof. Let $Q = \{v_1, \ldots, v_q\}$ be a quorum of minimal size in $S$, with sizes $|Q| = q$ and $|S| = n$. Let $Z$ be an access strategy for $S$. Every other quorum in $S$ intersects in at least one element with this quorum $Q$. Each time a quorum is accessed, at least one node in $Q$ is accessed as well, yielding a lower bound of $L_Z(v_i) \geq 1/q$ for some $v_i \in Q$.

Furthermore, as $Q$ is minimal, at least $q$ nodes need to be accessed, yielding $W_S^Z(q) \geq q$. Thus, $L_Z(v_i) \geq q/n$ for some $v_i \in Q$, as each time $q$ nodes are accessed, the load of the most accessed node is at least $q/n$.

Combining both ideas leads to $L_Z(S) \geq \max(1/q, q/n) \Rightarrow L_Z(S) \geq 1/\sqrt{n}$.

Thus, $L(S) \geq 1/\sqrt{n}$, as $Z$ can be any access strategy.

Remarks:
- Can we achieve this load?

15.2 Grid Quorum Systems

Definition 15.7 (Basic Grid quorum system). Assume $\sqrt{n} \in \mathbb{N}$, and arrange the $n$ nodes in a square matrix with side length of $\sqrt{n}$, i.e., in a grid. The basic Grid quorum system consists of $\sqrt{n}$ quorums, with each containing the full row $i$ and the full column $i$, for $1 \leq i \leq \sqrt{n}$.

Figure 15.8: The basic version of the Grid quorum system, where each quorum $Q_i$, with $1 \leq i \leq \sqrt{n}$ uses row $i$ and column $i$. The size of each quorum is $2\sqrt{n} - 1$ and two quorums overlap in exactly two nodes. Thus, when the access strategy $Z$ is uniform (i.e., the probability of each quorum is $1/\sqrt{n}$), the work is $2\sqrt{n} - 1$, and the load of every node is in $\Theta(1/\sqrt{n})$.

Remarks:
- Consider the right picture in Figure 15.8: The two quorums intersect in two nodes. If both quorums were to be accessed at the same time, it is not guaranteed that at least one quorum will lock all of its nodes, as they could enter a deadlock!
- In the case of just two quorums, one could solve this by letting the quorums just intersect in one node, see Figure 15.9. However, already with three quorums the same situation could occur again, progress is not guaranteed!
- However, by deviating from the ‘access all at once’ strategy, we can guarantee progress if the nodes are totally ordered!

Figure 15.9: There are other ways to choose quorums in the grid s.t. pairwise different quorums only intersect in one node. The size of each quorum is between $\sqrt{n}$ and $2\sqrt{n} - 1$, i.e., the work is in $\Theta(1/\sqrt{n})$. When the access strategy $Z$ is uniform, the load of every node is in $\Theta(1/\sqrt{n})$.

Algorithm 15.10 Sequential Locking Strategy for a Quorum $Q$

1. Attempt to lock the nodes one by one, ordered by their identifiers
2. Should a node be already locked, release all locks and start over

Theorem 15.11. If each quorum is accessed by Algorithm 15.10, at least one quorum will obtain a lock for all of its nodes.

Proof. We prove the theorem by contradiction. Assume no quorum can make progress, i.e., for every quorum we have: At least one of its nodes is locked by another quorum. Let $v$ be the node with the highest identifier that is locked by some quorum $Q$. Observe that $Q$ already locked all of its nodes with a smaller identifier than $v$, otherwise $Q$ would have restarted. As all nodes with a higher identifier than $v$ are not locked, $Q$ either has locked all of its nodes or can make progress – a contradiction. As the set of nodes is finite, one quorum will eventually be able to lock all of its nodes.

Remarks:
- But now we are back to sequential accesses in a distributed system? Let’s do it concurrently with the same idea, i.e., resolving conflicts by the ordering of the nodes. Then, a quorum that locked the highest identifier so far can always make progress!

Theorem 15.13. If the nodes and quorums use Algorithm 15.12, at least one quorum will obtain a lock for all of its nodes.
15.3. FAULT TOLERANCE

Algorithm 15.12 Concurrent Locking Strategy for a Quorum Q

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

1. repeat
2. Attempt to lock all nodes of the quorum Q
3. for each node v ∈ Q that was not able to be locked by Q do
   4. exchange vQ and vQ with the quorum Q' that locked v
   5. if vQ > vQ then
      6. Q' releases lock on v and Q acquires lock on v
   7. end if
8. end for
9. until all nodes of the quorum Q are locked

Proof. The proof is analogous to the proof of Theorem 15.11: Assume for contradiction that no quorum can make progress. However, at least the quorum with the highest vQ can always make progress – a contradiction! As the set of nodes is finite, at least one quorum will eventually be able to acquire a lock on all of its nodes. □

Remarks:
• What if a quorum locks all of its nodes and then crashes? Is the quorum system dead now? This issue can be prevented by, e.g., using leases instead of locks: leases have a timeout, i.e., a lock is released eventually.

15.3 Fault Tolerance

Definition 15.14 (resilience). If any f nodes from a quorum system S can fail s.t. there is still a quorum Q ∈ S without failed nodes, then S is f-resilient.

The largest such f is the resilience R(S).

Theorem 15.15. Let S be a Grid quorum system where each of the n quorums consists of a full row and a full column. S has a resilience of √n − 1.

Proof. If all √n nodes on the diagonal of the grid fail, then every quorum will have at least one failed node. Should less than √n nodes fail, then there is a row and a column without failed nodes. □

Remarks:
• The Grid quorum system in Theorem 15.15 is different from the Basic Grid quorum system described in Definition 15.7. In each quorum in the Basic Grid quorum system the row and column index are identical, while in the Grid quorum system of Theorem 15.15 this is not the case.

Definition 15.16 (failure probability). Assume that every node works with a fixed probability p (in the following we assume concrete values, e.g. p > 1/2).

The failure probability Fp(S) of a quorum system S is the probability that at least one node of every quorum fails.

Remarks:
• The asymptotic failure probability is Fp(S) for n → ∞.

Facts 15.17. A version of a Chernoff bound states the following:
Let x1, . . . , xh be independent Bernoulli-distributed random variables with Pr[xi = 1] = p and Pr[xi = 0] = 1 − p = q, then for X := h i=1 xi and µ := h i=1 µq the following holds:

\[ Pr[|X − µ| > δ] ≤ e^{−δ^2/2} \]

for all 0 < δ < 1. Pr[X ≤ (1 − δ)µ] ≤ e−eh(µ−1)2.

Theorem 15.18. The asymptotic failure probability of the Majority quorum system is 0.

Proof. In a Majority quorum system each quorum contains exactly ⌈ n/2 ⌉ + 1 nodes and each subset of nodes with cardinality ⌈ n/2 ⌉ + 1 forms a quorum. The Majority quorum system fails, if only ⌈ n/2 ⌉ nodes work. Otherwise there is at least one quorum available. In order to calculate the failure probability we define the following random variables:

\[ x_i = \begin{cases} 1, & \text{if node i works, happens with probability } p \\ 0, & \text{if node i fails, happens with probability } q = 1 - p \end{cases} \]

and \( X := \sum_{i=1}^{n} x_i \) with \( µ = np \), whereas \( X \) corresponds to the number of working nodes. To estimate the probability that the number of working nodes is less than \( ⌈ n/2 ⌉ + 1 \) we will make use of the Chernoff inequality from above. By setting \( δ = 1 - \frac{1}{2} \) we obtain

\[ F_p(S) = Pr[X < ⌈ n/2 ⌉] ≤ Pr[X ≤ ⌈ n/2 ⌉ − 1] = Pr[X ≤ (1 − δ)µ] \]

With \( δ = 1 - \frac{1}{2} \) we have \( 0 < δ ≤ 1/2 \) due to 1/2 < p ≤ 1. Thus, we can use the Chernoff bound and get \( F_p(S) ≤ e^{-e(1/2)^2} ≤ e^{-e(δ^2)} \).

Theorem 15.19. The asymptotic failure probability of the Grid quorum system is 1.

Proof. Consider the n = d · d nodes to be arranged in a d × d grid. A quorum always contains one full row. In this estimation we will make use of the Bernoulli inequality which states that for all \( n \in \mathbb{N}, x ≥ 1 : (1 + x)^n ≥ 1 + nx \).

The system fails, if in each row at least one node fails (which happens with probability 1 − pδ) for a particular row, as all nodes work with probability pδ. Therefore we can bound the failure probability from below with:

\[ F_p(S) ≥ Pr[\text{at least one failure per row}] = (1 − pδ)^d ≥ 1 − dpδ \text{ → 1}. \]

Remarks:
• Now we have a quorum system with optimal load (the Grid) and one with fault-tolerance (Majority), but what if we want both?

Definition 15.20 (B-Grid quorum system). Consider n = dh nodes, arranged in a rectangular grid with h · r rows and d columns. Each group of r rows is a band, and r columns in a column restricted to a band are called a mini-column. A quorum consists of one mini-column in every band and one element from each mini-column of one band; thus every quorum has d + hr − 1 elements. The B-Grid quorum system consists of all such quorums.
15.4 Byzantine Quorum Systems

While failed nodes are bad, they are still easy to deal with: just access another quorum where all nodes can respond! Byzantine nodes make life more difficult however, as they can pretend to be a regular node, i.e., one needs more sophisticated methods to deal with them. We need to ensure that the intersection of two quorums always contains a non-byzantine (correct) node and furthermore, the byzantine nodes should not be allowed to infiltrate every quorom. In this section we study three counter-measures of increasing strength, and their implications on the load of quorum systems.

Definition 15.24 (f-disseminating). A quorum system $S$ is $f$-disseminating if (1) the intersection of two different quorums always contains $f + 1$ nodes, and (2) for any set of $f$ byzantine nodes, there is at least one quorum without byzantine nodes.

Remarks:

- Thanks to (2), even with $f$ byzantine nodes, the byzantine nodes cannot stop all quorums by just pretending to have crashed. At least one quorum will survive. We will also keep this assumption for the upcoming more advanced byzantine quorum systems.
- Byzantine nodes can also do something worse than crashing - they could falsify data! Nonetheless, due to (1), there is at least one non-byzantine node in every quorum intersection. If the data is self-verifying by, e.g., authentication, then this one node is enough.
- If the data is not self-verifying, then we need another mechanism.

Definition 15.25 ($f$-masking). A quorum system $S$ is $f$-masking if (1) the intersection of two different quorums always contains $2f + 1$ nodes, and (2) for any set of $f$ byzantine nodes, there is at least one quorum without byzantine nodes.

Remarks:

- Note that except for the second condition, an $f$-masking quorum system is the same as a $2f$-disseminating system. The idea is that the non-byzantine nodes (at least $f + 1$ can outvote the byzantine ones (at most $f$), but only if all non-byzantine nodes are up-to-date!
- This raises an issue not covered yet in this chapter. If we access some quorum and update its values, this change still has to be disseminated to the other nodes in the byzantine quorum system.Opaque quorum systems deal with this issue, which are discussed at the end of this section.

Table 15.23 Overview of the different quorum systems regarding resilience, work, load, and their asymptotic failure probability. The best entries in each row are set in bold.

<table>
<thead>
<tr>
<th></th>
<th>Singleton Majority Grid B-Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>$f &gt; n/2$</td>
</tr>
<tr>
<td>Load</td>
<td>$1 &gt; 1/2$</td>
</tr>
<tr>
<td>Resilience</td>
<td>$0 &lt; n/2$</td>
</tr>
<tr>
<td>$P.\text{Prob.}$</td>
<td>$1 - p$</td>
</tr>
</tbody>
</table>

* Setting $d = \sqrt{n}$ and $r = \log n$

** Assuming prob. $q = (1 - p)$ is constant but significantly less than $1/2$
• $f$-disseminating quorum systems need more than $3f$ nodes and $f$-
masking quorum systems need more than $4f$ nodes. Essentially, the quorums may not contain too many nodes, and the different intersec-
tion properties lead to the different bounds.

Theorem 15.26. Let $S$ be a $f$-disseminating quorum system. Then $L(S) \geq \sqrt{(f + 1)/n}$ holds.

Theorem 15.27. Let $S$ be a $f$-masking quorum system. Then $L(S) \geq \sqrt{(2f + 1)/n}$ holds.

Proofs of Theorems 15.26 and 15.27. The proofs follow the proof of Theorem 15.6, by observing that now not just one element is accessed from a minimal quorum, but $f + 1$ or $2f + 1$, respectively.

Definition 15.28 ($f$-masking Grid quorum system). A $f$-masking Grid quo-
rum system is constructed as the grid quorum system, but each quorum contains
one full column and $f + 1$ rows of nodes, with $2f + 1 \leq \sqrt{n}$.

Figure 15.29: An example how to choose a quorum in the $f$-masking Grid quorum system. Then, $L(S) \geq \sqrt{(2f + 1)/n}$ holds.

Remarks:
• The $f$-masking Grid nearly hits the lower bound for the load of $f$-
masking quorum systems, but not quite. A small change and we will be
optimal asymptotically.

Definition 15.30 ($M$-Grid quorum system). The $M$-Grid quorum system is constructed as the grid quorum as well, but each quorum contains $\sqrt{f + 1}$ rows and $\sqrt{f + 1}$ columns of nodes, with $f \leq \frac{\sqrt{n}}{2}$.

Figure 15.31: An example how to choose a quorum in the $M$-Grid with $f = 3$, i.e., 2 rows and 2 columns. The load is in $\Theta(\sqrt{f/n})$ when the access strategy is chosen to be uniform. Two quorums overlap with each row intersecting each other’s column, i.e., $2\sqrt{f + 1} = 2f + 2$ nodes.

Corollary 15.32. The $f$-masking Grid quorum system and the $M$-Grid quorum system are $f$-masking quorum systems.

Remarks:
• We achieved nearly the same load as without byzantine nodes! How-
ever, as mentioned earlier, what happens if we access a quorum that is
not up-to-date, except for the intersection with an up-to-date quorum? Surely we can fix that as well without too much loss!
• This property will be handled in the last part of this chapter by opaque
quorum systems. It will ensure that the number of correct up-to-date
nodes accessed will be larger than the number of out-of-date nodes
combined with the byzantine nodes in the quorum (cf. (15.33.1)).

Definition 15.33 ($f$-opaque quorum system). A quorum system $S$ is $f$-opa-
que if the following two properties hold for any set of $f$ byzantine nodes $F$ and any
two different quorums $Q_1, Q_2$:

\begin{align*}
|Q_1 \cap Q_2| &> \frac{|Q_2 \cap F| + |Q_1 |}{2} & (15.33.1) \\
(F \cap Q) &\neq \emptyset \text{ for some } Q \in S & (15.33.2)
\end{align*}

Theorem 15.35. Let $S$ be a $f$-opaque quorum system. Then, $n > 5f$.

Proof. Due to (15.33.2), there exists a quorum $Q_1$ with size at most $n - f$. With (15.33.1), $|Q_1| \geq f$ holds. Let $F_1$ be a set of $f$ (byzantine) nodes $F_1 \subseteq Q_1$, and with (15.33.2), there exists a $Q_2 \subseteq V \setminus F_1$. Thus, $|Q_1 \cap Q_2| \leq n - 2f$. With (15.33.1), $|Q_1 \cap Q_2| > f$ holds. Thus, one could choose $f$ (byzantine) nodes $F_2$ with $F_2 \subseteq (Q_1 \cap Q_2)$. Using (15.33.1) one can bound $n - 3f$ from below:

$n - 3f > |Q_2 \cap Q_1| - |F_2| \geq |(Q_2 \cap Q_1) \setminus (Q_1 \cap F_2)| \geq |F_1| + |F_2| = 2f$. \hfill \Q
BYZANTINE QUORUM SYSTEMS

The f-masking Grid quorum system and opaque quorum systems are from [MR98], and the M-Grid quorum system was introduced in [MRW97]. Both papers also mark the start of the formal study of Byzantine quorum systems. The f-masking and the M-Grid have asymptotic failure probabilities of 1, more complex systems with better values can be found in these papers as well.

Quorum systems have also been extended to cope with nodes dynamically leaving and joining, see, e.g., the dynamic paths quorum system in [NW05].

For a further overview on quorum systems, we refer to the book by Vukolić [Vuk12] and the article by Merideth and Reiter [MR10].

This chapter was written in collaboration with Klaus-Tycho Förster.

Bibliography


