# Consistency Analysis of Replication-Based Probabilistic Key-Value Stores

Ramy E. Ali

***Abstract*—Partial quorum systems are widely used in distributed key-value stores due to their latency benefits at the expense of providing weaker consistency guarantees. The probabilistically bounded staleness framework (PBS) studied the latency-consistency trade-off of Dynamo-style partial quorum systems through Monte Carlo event-based simulations. In this paper, we study the latency-consistency trade-off for such systems analytically and derive a closed-form expression for the inconsistency probability. Our approach allows fine-tuning of latency and consistency guarantees in key-value stores, which is intractable using Monte Carlo event-based simulations.**

***Index Terms*—Eventual Consistency, Probabilistic Consistency, Partial Quorums.**

## I. Introduction

Key-value stores[1] are essential for many applications such as reservation systems, financial transactions and distributed computing. Such systems commonly replicate the data across multiple servers to make the data available and accessible with low latency despite the possible failures and stragglers. In these systems, the data is frequently updated and it is desirable to make the latest version of the data accessible by the different users. This requirement is known as consistency in distributed systems [1]. In order to ensure strong consistency, these systems use strict quorums where the write and the read quorums must intersect [1], [2]. Specifically, in a system of $N$ servers, where $W$ and $R$ denote the write and the read quorum sizes respectively, $W$ and $R$ are chosen such that $W + R > N$. In order to have fast access to the data that is critical for many applications, many key-values stores including Amazon's Dynamo [3] and Cassandra [4] allow non-strict (partial, probabilistic or sloppy) quorums where $W + R \leq N$. These systems however only guarantee that the users will eventually return the latest version of the data if there are no new write operations [5], [6]. However, eventual consistency does not specify how fast this will happen.

Several works studied probabilistic quorum systems, attempted to quantify the staleness of the data retrieved, how soon users can retrieve consistent data and providing adaptive consistency guarantees depending on the application including [7]–[20]. In [7], $\epsilon$-intersecting probabilistic quorum systems were designed such that the probability that any two quorums do not intersect is at most $\epsilon$. In [12], an adaptive approach was proposed that tunes the inconsistency probability, assuming that the response time of the servers are neglected, through controlling the number of servers involved in the read operations at the run-time based on a monitoring module. The monitoring module provides a real-time estimate of the network delays. In this approach, the write operation completes when any server responds to the write client. While the data is being propagated to the remaining servers, any server is pessimistically considered stale except the first server that responded to the write operation. Hence, this approach does not fully capture expanding write quorums (anti-entropy) [21].

In [11], [14], the trade-off that partial quorum systems provide between the staleness of the retrieved data and the latency was studied in 3-way replication-based key-value stores. Specifically, this work answered the question of how stale is the retrieved data through the notion of $l$-staleness, which measures the probability that the users retrieve one of the $l$ latest complete versions. The question of how eventual a user can read consistent data is also studied in [11], [14] through the notion of $t$-visibility that measures the probability of returning the value of a write operation $t$ units of times after it completes. While the write operation completes upon receiving acknowledgments from any $W$ servers, more servers receive the write request after that and the write quorum can continue to expand. Characterizing the $t$-visibility is challenging as it depends on how the write quorum expands based on the delays of the write and the read requests. Hence, the study of [14] focused on obtaining insights about this question for 3-way replication through Monte Carlo simulations.

In this paper, we study the problem of providing probabilistic guarantees for partial quorum systems analytically for replication-based key-value stores. We study the inconsistency probability for such systems in terms of the quorum sizes, mean write and read delays. For 3-way replication-based systems, we derive an explicit simple closed-form expression for the inconsistency probability in terms of those parameters.

The rest of this paper is organized as follows. In Section II, we describe the system model and provide a background. In Section III, we study expanding quorums that have a dynamic size. We analyze the inconsistency probability of replication-based partial quorum systems in Section IV. Finally, concluding remarks are discussed in Section V.

## II. System Model and Background

In this section, we describe our system model and provide a brief background about the order statistics and the sum of exponential random variables.

Ramy E. Ali (E-mail: ramy.ali@psu.edu) is with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, PA, and was with Bell Labs, NJ.

[1]Key-value stores are shared databases that store the data as a collection of key-value pairs.

### A. System Model: Partial Quorums

We consider a distributed system with $N$ servers denoted by $\mathcal{N} = \{1, 2, \cdots, N\}$ storing a shared object. A client that issues a write request sends the request to all servers and waits for the acknowledgment of $W$ servers for the write operation to complete. We denote the time that a write request takes to reach to server $i$ in addition to the server's response time by $X_i$, where $i \in \mathcal{N}$. We assume that $X_1, X_2, \cdots, X_N$ are independent and identically distributed exponential random variables with parameter $\lambda$. A client that issues a read request sends the request to all servers and waits for $R$ servers to respond. The time the read request takes to reach server $i$ and the server's response time is denoted by $Z_i, i \in \mathcal{N}$. We assume that the read delays $Z_1, Z_2, \cdots, Z_N$ are independent and identically distributed random variables according to exponential distribution with parameter $\xi$. Finally, we assume that write and read acknowledgments are instantaneous (See Remark 1).

In strict quorum systems, $W$ and $R$ are chosen such that $W+R > N$. In partial quorum systems however, $W+R \leq N$ and hence the write and the read quorums may not intersect. This may result in a consistency violation. In real-world quorum systems however, the write quorum expands as the write request propagate to more servers. In [11], the notion of $t$-visibility was developed which aims to capture the probability of inconsistency for expanding quorums for a read operation that starts $t$ units of time after the write completes. Our goal in this work is to characterize the inconsistency probability for expanding quorums as a function of $t$ and the quorum sizes.

**Remark 1.** While we assume that the write acknowledgments are instantaneous for simplicity, a deterministic delay of the acknowledgment denoted by $d$ can be taken into account by studying the consistency $t + d$ units of time after $W$ servers respond to the write request.

### B. Background: Order Statistics and Sum of Exponentials

In this subsection, we provide a brief background about exponential random variables that we build on later in Section III to study expanding quorums. We first recall the following useful Lemma [22] for the order statistics of independent exponential random variables with a common parameter $\lambda$.

**Lemma 1** (Order Statistics of Independent Exponentials)**.** Let $X_1, X_2, \cdots, X_n$ be independent and identically distributed random variables according to $\exp(\lambda)$, then we have

$$Y_i \coloneqq X_{(i)} - X_{(i-1)} \sim \exp((n-i+1)\lambda), \tag{1}$$

where $X_{(i)}$ denotes the $i$-th smallest of $X_1, X_2, \cdots, X_n$, $i \in \{1, 2, \cdots, n\}$ and $X_{(0)} = 0$.

We also recall the following Lemma from [23] which studies the sum of independent exponential random variables with different parameters.

**Lemma 2** (Sum of Exponentials)**.** Let $Y_1, Y_2, \cdots, Y_n$ be independent exponentials random variables with parameters $\lambda_1, \lambda_2, \cdots, \lambda_n$ respectively, where $f_{Y_i}(y)$ denotes the density function of $Y_i$. The density function of

$$Z \coloneqq \sum_{i=1}^{n} Y_i \tag{2}$$

is given by

$$f_Z(z) = \sum_{i=1}^{n} f_i(z) \prod_{\substack{j=1,\\ j\neq i}}^{n} \frac{\lambda_j}{\lambda_j - \lambda_i}, \ z \geq 0. \tag{3}$$

## III. Expanding Quorums

In this section, we characterize the probability distribution of the number of servers in the write quorum $t$ units of time after the write completes. As we have explained, a client that issues a write request sends the request to all $N$ servers and waits to receive acknowledgments from any $W$ servers. The first $W$ received responses determine the write latency $X_{(W)}$, but the write quorum will continue to expand as more servers receive the write request. We denote the set of servers that have received the write value $t$ units of time after it completes by $\mathcal{S}(t)$, where $S(t) \coloneqq |\mathcal{S}(t)|$ and $S(0) = W$. In Theorem 1, we characterize the probability mass function (PMF) of $S(t)$.

**Theorem 1** (Dynamic Quorum Size)**.** The PMF of the number of servers that have received a complete version $t$ units of time after it completes, $S(t)$, is given by

$$\Pr[S(t) = W] = e^{-\lambda_{W+1} t}, \tag{4}$$

$$\Pr[S(t) = s] = \sum_{i=W+1}^{s+1} (-1)^{s-i} (1 - e^{-\lambda_i t}) \binom{N-W}{N-i+1} \binom{N-i+1}{s-i+1}, \tag{5}$$

for $s \in \{W+1, W+2, \cdots, N\}$, where $\lambda_i = (N-i+1)\lambda$.

We provide the proof of Theorem 1 in Appendix A.
In Fig. 1, we show the PMF of $S(1)$ for $N = 3, W = 1$ and $\lambda = 1$. In Fig. 2, we show the PMF of $S(1)$ for $N = 3, W = 2$ and $\lambda = 1$.

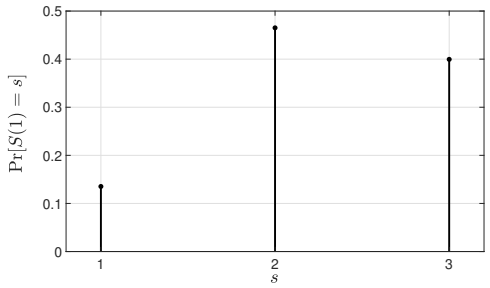


Fig. 1: The probability mass function of $S(1)$ for the case where $N = 3, W = 1$ and $\lambda = 1$.

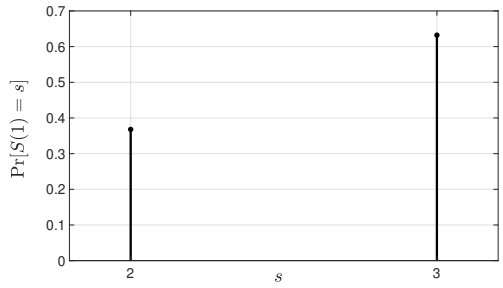


Fig. 2: The probability mass function of $S(1)$ for the case where $N = 3, W = 2$ and $\lambda = 1$.

## IV. Consistency Analysis

In this section, we study the inconsistency probability of replication-based partial quorum systems. The worst-case probability of inconsistency assuming non-expanding write quorums and instantaneous reads is given by

$$p = \frac{\binom{N-W}{R}}{\binom{N}{R}}. \tag{6}$$

Since write quorum expands as the write request propagate to more servers, equation (6) is in fact an upper bound of the inconsistency probability [11].

Our objective in this section is to characterize the exact inconsistency probability for expanding quorums. The read client returns inconsistent data if the first $R$ servers that respond to the read request return stale data. A server is considered stale if it replies to the read request before receiving the latest compete version. That is, server $i$ is stale if $X_{(W)} + t + Z_i < X_i$. Denote the first $R$ servers that respond to the read request by $\mathcal{R} = \{r_1, r_2, \cdots, r_R\}$, where $r_1$ is the server the replies first, $r_2$ is the server that replies second and so on. The event that server $r_j$ is stale is expressed as follows

$$\begin{aligned} E_j &= \{X_{(W)} + t + Z_{(j)} < X_{r_j}\} \\ &= \{r_j \notin \mathcal{S}(t + Z_{(j)})\}. \end{aligned} \tag{7}$$

where $j \in \mathcal{R}$. In order to keep the notation simple, we denote $\mathcal{S}(t + Z_{(j)})$ by $\mathcal{S}_j$. The probability that a read returns stale data $t$ units of time after that latest version completes is the probability that all servers in $\mathcal{R}$ return stale data. Thus, the inconsistency probability can be expressed as follows

$$\begin{aligned} p_t &= \Pr[\text{All servers in } \mathcal{R} \text{ are stale}] \\ &= \Pr\left[\bigcap_{j=1}^{R} E_j\right]. \end{aligned} \tag{8}$$

We note that characterizing the inconsistency probability exactly is challenging as $E_1, E_2, \cdots, E_R$ are dependent, hence we express the inconsistency probability as follows

$$\begin{aligned} p_t &= \Pr\left[r_1 \notin \mathcal{S}_1, r_2 \notin \mathcal{S}_2, \cdots, r_R \notin \mathcal{S}_R\right] \\ &= \Pr\left[r_R \notin \mathcal{S}_R | r_{R-1} \notin \mathcal{S}_{R-1}, \cdots, r_1 \notin \mathcal{S}_1\right] \cdots \\ &\quad \Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \notin \mathcal{S}_1\right] \Pr\left[r_1 \notin \mathcal{S}_1\right]. \end{aligned} \tag{9}$$

In order to find the inconsistency probability, we first need to characterize the PMF of the number of servers in the write quorum $t + Z_{(j)}$ units of time after the write completes.

**Lemma 3.** The probability mass function of the number of servers in the write quorum $t + Z_{(j)}$ units of time, where $j \in \mathcal{R}$, after the write completes is given by

$$\begin{aligned} &\Pr[S(t + Z_{(j)}) = W] = e^{-\lambda_{W+1} t} \\ &\qquad \sum_{l=1}^{j} \binom{N}{j} \binom{j}{l} \frac{(-1)^{j-l} \xi_{N-l+1}}{\xi_l + \lambda_{W+1}}, \end{aligned} \tag{10}$$

$$\begin{aligned} &\Pr[S(t + Z_{(j)}) = s] = \\ &\sum_{i=W+1}^{s+1} (-1)^{s-i} \binom{N-W}{N-i+1} \binom{N-i+1}{s-i+1} \\ &\left(1 - e^{-\lambda_i t} \sum_{l=1}^{j} \binom{N}{j} \binom{j}{l} \frac{(-1)^{j-l} \xi_{N-l+1}}{\xi_l + \lambda_i}\right), \end{aligned} \tag{11}$$

for $s \in \{W+1, \cdots, N\}$, where $\xi_j = (N - j + 1)\xi$ and $\lambda_j = (N - j + 1)\lambda$.

The proof of Lemma 3 is straightforward, but we provide it in Appendix B for completeness.

In Theorem 2, we provide our main result in which we characterize the inconsistency probability of the widely-used 3-way replication technique.

**Theorem 2** (Inconsistency Probability of Replication-based Systems with $N = 3$)**.**

- The worst-case inconsistency probability for the case where $W = 1$ and $R = 1$ is expressed as follows

$$p_t = \frac{2\xi e^{-\lambda t}}{\lambda + 3\xi}. \tag{12}$$

- The worst-case inconsistency probability for the case where $W = 2$ and $R = 1$ is expressed as follows

$$p_t = \frac{\xi e^{-\lambda t}}{\lambda + 3\xi}. \tag{13}$$

- The worst-case inconsistency probability for the case where $W = 1$ and $R = 2$ is expressed as follows

$$\begin{aligned} p_t = &\frac{6\xi^3 e^{-2\lambda t}}{(\lambda + 2\xi)(\lambda + 3\xi)} \\ &\left(\frac{2\lambda}{(\lambda + 2\xi)(\lambda + 3\xi)} - \frac{(\lambda - \xi)e^{-\lambda t}}{(\lambda + \xi)(2\lambda + 3\xi)}\right). \end{aligned} \tag{14}$$

The proof of Theorem 2 can be found in Appendix C.

**Remark 2.** It can be verified that at $t = 0$, the limit of the inconsistency probability of Theorem 2 as $\xi$ grows is equal to the inconsistency probability assuming instantaneous reads given in (6). That is, we have

$$\lim_{\xi \to \infty} p_0 = p. \tag{15}$$

It is worth noting that the upper bound of the inconsistency probability given in (6) is quite loose. In order to see this, we

observe that this bound gives an inconsistency probability of $1/3$ for the case where $W = 2, R = 1$ and also for the case where $W = 1, R = 2$. Hence, this bound does not differentiate between these two cases.

We show the probability of inconsistency for the different cases in Fig. 3 as a function of $t$.

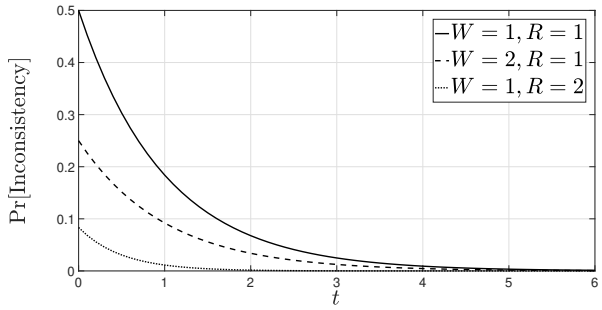


Fig. 3: The probability of inconsistency for the case where $N = 3, \lambda = 1$ and $\xi = 1$.

**Remark 3** (Asymmetry)**.** It is worth noting that the inconsistency probability is asymmetric in the write and read quorum sizes and also the write and read mean delays.

**Remark 4** (Replication Factor)**.** While the case of $N = 3$ is the typical case in replication-based systems, our approach can also be used to derive the inconsistency probability for general $N, W$ and $R$ [24, Ch. 4]. In general, there are $R!$ cases to be considered. For instance, for $R = 3$, the following cases lead to violating the consistency

1) $(r_3 \notin \mathcal{S}_3, r_2 \in \mathcal{S}_3 - \mathcal{S}_2, r_1 \in \mathcal{S}_2 - \mathcal{S}_1)$,
2) $(r_3 \notin \mathcal{S}_3, r_2 \notin \mathcal{S}_3, r_1 \in \mathcal{S}_2 - \mathcal{S}_1)$,
3) $(r_3 \notin \mathcal{S}_3, r_2 \in \mathcal{S}_3 - \mathcal{S}_2, r_1 \in \mathcal{S}_3 - \mathcal{S}_2)$,
4) $(r_3 \notin \mathcal{S}_3, r_2 \notin \mathcal{S}_3, r_1 \in \mathcal{S}_3 - \mathcal{S}_2)$,
5) $(r_3 \notin \mathcal{S}_3, r_2 \in \mathcal{S}_3 - \mathcal{S}_2, r_1 \notin \mathcal{S}_3)$,
6) $(r_3 \notin \mathcal{S}_3, r_2 \notin \mathcal{S}_3, r_1 \notin \mathcal{S}_3)$.

**Remark 5** (Beyond Replication and Exponential Delays)**.** The proof technique of Theorem 2 can be used to characterize the inconsistency probability for any given distributions of the write and read delays such as shifted exponential distribution. This approach can be extended also to simple erasure-coded probabilistic key-value stores, where each version of the data is encoded using a maximum distance separable (MDS) code of dimension $k$. In erasure-coded strict quorum systems, $W$ and $R$ are chosen such that $W+R-N \geq k$ to ensure that the write and read quorums intersect in at least $k$ servers. In erasure-coded probabilistic quorum systems however, the quorum sizes may be selected such that $W + R - N < k$ to provide faster access to the data. Characterizing the inconsistency probability of these systems is challenging and we refer the reader to [24, Ch. 5] for a follow up in this direction.

## V. Conclusion

In this paper, we have studied the consistency-latency tradeoff for Dynamo-style replication-based key-value stores analytically and derived a closed-form expression for the inconsistency probability for the 3-way replication technique. Our study allows fine-tuning of latency and consistency guarantees based on the mean values of the write and read delays of the data store. An immediate future work is to incorporate our tuning policy in a distributed key-value store and evaluate its performance. Extending this study to derive a tight upper bound on the inconsistency probability for any given distributions of the write delays, read delays and acknowledgments delays are also interesting future research directions.

## VI. Appendices

### A. Proof of Theorem 1

For the case where $s = W$, we have

$$\begin{aligned}\Pr[S(t) = W] &= \Pr[S(t) \leq W] \\ &= \Pr[X_{(W+1)} - X_{(W)} > t] \\ &= e^{-\lambda_{W+1} t},\end{aligned}$$

where the last equality follows Lemma 1.
For the case were $s \in \{W+1, W+2, \cdots, N\}$, we have

$$\begin{aligned}\Pr[S(t) = s] &= \Pr[S(t) \leq s] - \Pr[S(t) \leq s-1] \\ &= \Pr[X_{(s+1)} - X_{(W)} > t] - \Pr[X_{(s)} - X_{(W)} > t] \\ &= \Pr[X_{(s)} - X_{(W)} \leq t] - \Pr[X_{(s+1)} - X_{(W)} \leq t] \\ &= \Pr\left[\sum_{i=W+1}^{s} X_{(i)} - X_{(i-1)} \leq t\right] \\ &\quad - \Pr\left[\sum_{i=W+1}^{s+1} X_{(i)} - X_{(i-1)} \leq t\right] \\ &= \Pr\left[\sum_{i=W+1}^{s} Y_i \leq t\right] - \Pr\left[\sum_{i=W+1}^{s+1} Y_i \leq t\right],\end{aligned}$$

where $Y_i = X_{(i)} - X_{(i-1)}$. Since $X_1, X_2, \cdots, X_N$ are independent and identical exponential random variables, then $Y_i$ is an exponential random variable with parameter $\lambda_i = (N-i+1)\lambda$, where $i \in \{2, 3, \cdots, N\}$ from Lemma 1. Since $Y_1, Y_2, \cdots, Y_N$ are independent exponential random variables, from Lemma 2, we have

$$\begin{aligned}\Pr[S(t) = s] &= \Pr\left[\sum_{i=W+1}^{s} Y_i \leq t\right] - \Pr\left[\sum_{i=W+1}^{s+1} Y_i \leq t\right] \\ &= \sum_{i=W+1}^{s} F_i(t) \prod_{\substack{j=W+1, \\ j\neq i}}^{s} \frac{\lambda_j}{\lambda_j - \lambda_i} \\ &\quad - \sum_{i=W+1}^{s+1} F_i(t) \prod_{\substack{j=W+1, \\ j\neq i}}^{s+1} \frac{\lambda_j}{\lambda_j - \lambda_i} \\ &= \sum_{i=W+1}^{s} F_i(t) \frac{\lambda_i}{\lambda_i - \lambda_{s+1}} \prod_{\substack{j=W+1, \\ j\neq i}}^{s} \frac{\lambda_j}{\lambda_j - \lambda_i} \\ &\quad - F_{s+1}(t) \prod_{j=W+1}^{s} \frac{\lambda_j}{\lambda_j - \lambda_{s+1}}\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=W+1}^{s} F_i(t)\frac{N-i+1}{s-i+1}\prod_{\substack{j=W+1,\\ j\neq i}}^{s}\frac{N-j+1}{i-j} \\
&- F_{s+1}(t)\prod_{j=W}^{s-1}\frac{N-j}{s-j} \\
&= \sum_{i=W+1}^{s} F_i(t)\frac{N-i+1}{s-i+1}\prod_{\substack{j=W+1,\\ j\neq i}}^{s}\frac{N-j+1}{i-j} \\
&- F_{s+1}(t)\binom{N-W}{N-s} \\
&= \sum_{i=W+1}^{s} (1-e^{-\lambda_i t})\frac{N-i+1}{s-i+1}\prod_{\substack{j=W+1,\\ j\neq i}}^{s}\frac{N-j+1}{i-j} \\
&- (1-e^{-\lambda_{s+1} t})\binom{N-W}{N-s} \\
&= \sum_{i=W+1}^{s+1} (-1)^{s-i}(1-e^{-\lambda_i t}) \\
&\quad \binom{N-W}{N-i+1}\binom{N-i+1}{s-i+1}.
\end{aligned}
$$

### B. Proof of Lemma 3

Based on Lemma 2, we can express the probability density function of

$$
Z_{(j)} = \sum_{l=1}^{j} Z_{(l)} - Z_{(l-1)} \tag{16}
$$

as follows

$$
\begin{aligned}
f_{Z_{(j)}}(z) &= \sum_{l=1}^{j} f_l(z)\prod_{\substack{i=1,\\ i\neq l}}^{j}\frac{\xi_i}{\xi_i-\xi_l} \\
&= \sum_{l=1}^{j}(-1)^{j-l}\xi_{N-l+1}\binom{N}{j}\binom{j}{l}e^{-\xi_l z},
\end{aligned}
$$

where $z \geq 0$. Therefore, from Theorem 1, we can express $\Pr[S(t+Z_{(j)}) = W]$ as follows

$$
\begin{aligned}
\Pr[S(t+Z_{(j)}) = W] &= \int_0^\infty e^{-\lambda_{W+1}(t+z)} f_{Z(j)}(z)\, dz \\
&= e^{-\lambda_{W+1}t}\sum_{l=1}^{j}\binom{N}{j}\binom{j}{l}\frac{(-1)^{j-l}\xi_{N-l+1}}{\xi_l+\lambda_{W+1}}.
\end{aligned}
$$

where $\xi_j = (N-j+1)\xi$ and $\lambda_j = (N-j+1)\lambda$. Similarly for $s \in \{W+1, W+2, \cdots, N\}$, we have

$$
\begin{aligned}
\Pr[S(t+Z_{(j)}) = s] = &\sum_{i=W+1}^{s+1}(-1)^{s-i} \\
&\binom{N-W}{N-i+1}\binom{N-i+1}{s-i+1} \\
&\left(1-e^{-\lambda_i t}\sum_{l=1}^{j}\binom{N}{j}\binom{j}{l}\frac{(-1)^{j-l}\xi_{N-l+1}}{\xi_l+\lambda_i}\right).
\end{aligned}
$$

### C. Proof of Theorem 2

The probability of inconsistency for the case where $W = 1$ and $R = 1$ can be expressed as follows

$$
\begin{aligned}
p_t &= \Pr[r_1 \notin \mathcal{S}_1] \\
&= \sum_{s=W}^{N}\Pr[r_1 \notin \mathcal{S}_1 | S(t+Z_{(1)}) = s] \\
&\quad \Pr[S(t+Z_{(1)}) = s] \\
&= \sum_{s=W}^{N}\left(1-\frac{s}{N}\right)\Pr[S(t+Z_{(1)}) = s] \\
&= \frac{2}{3}\Pr[S(t+Z_{(1)}) = 1] + \frac{1}{3}\Pr[S(t+Z_{(1)}) = 2] \\
&= \frac{2}{3}\frac{\xi_1 e^{-2\lambda t}}{\xi_1+2\lambda} + \frac{1}{3}\left(\frac{2\xi_1 e^{-\lambda t}}{\xi_1+\lambda} - \frac{2\xi e^{-2\lambda t}}{\xi_1+2\lambda}\right) \\
&= \frac{2}{3}\frac{\xi_1 e^{-\lambda t}}{\xi_1+\lambda} = \frac{2\xi e^{-\lambda t}}{3\xi+\lambda}.
\end{aligned}
$$

Similarly, for the case where $W = 2$ and $R = 1$, we have

$$
p_t = \frac{1}{3}\Pr[S(t+Z_{(1)}) = 2] = \frac{1}{3}\frac{\xi_1 e^{-\lambda t}}{\xi_1+\lambda} = \frac{\xi e^{-\lambda t}}{3\xi+\lambda}.
$$

For the case where $R = 2$, we can express the probability of inconsistency as follows

$$
\begin{aligned}
p_t &= \Pr\left[r_2 \notin \mathcal{S}_2, r_1 \notin \mathcal{S}_1\right] \\
&= \Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \notin \mathcal{S}_1\right]\Pr\left[r_1 \notin \mathcal{S}_1\right].
\end{aligned}
$$

If $r_1 \notin \mathcal{S}_1$, it may happen that $r_1 \notin \mathcal{S}_2$ as well or $r_1 \in \mathcal{S}_2$ and these two cases need to be handled separately. Therefore, we express the inconsistency probability as follows

$$
\begin{aligned}
p_t &= \Pr\left[r_2 \notin \mathcal{S}_2, r_1 \notin \mathcal{S}_1\right] \\
&= \Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \notin \mathcal{S}_2\right]\Pr\left[r_1 \notin \mathcal{S}_2\right] \\
&+ \Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \in \mathcal{S}_2 - \mathcal{S}_1\right]\Pr[r_1 \in \mathcal{S}_2 - \mathcal{S}_1].
\end{aligned}
$$

It is important to note that

$$
\Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \in \mathcal{S}_2 - \mathcal{S}_1\right] = \Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \in \mathcal{S}_2\right].
$$

Hence, we have

$$
\begin{aligned}
p_t &= \Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \notin \mathcal{S}_2\right]\Pr\left[r_1 \notin \mathcal{S}_2\right] \\
&+ \Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \in \mathcal{S}_2\right](\Pr[r_1 \in \mathcal{S}_2] - \Pr[r_1 \in \mathcal{S}_1]),
\end{aligned}
$$

where

$$
\begin{aligned}
\Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \notin \mathcal{S}_2\right] &= \sum_{s=W}^{N-1}\left(1-\frac{s}{N-1}\right) \\
&\quad \Pr\left[S(t+Z_{(2)}) = s\right] \\
&= \frac{1}{2}\Pr\left[S(t+Z_{(2)}) = 1\right], \qquad (17)
\end{aligned}
$$

$$
\begin{aligned}
\Pr\left[r_2 \notin \mathcal{S}_2 | r_1 \in \mathcal{S}_2\right] &= \sum_{s=W}^{N}\left(1-\frac{s-1}{N-1}\right) \\
&\quad \Pr\left[S(t+Z_{(2)}) = s\right]
\end{aligned}
$$

$$= \frac{1}{2} \Pr\left[S(t + Z_{(2)}) = 2\right], \quad (18)$$

$$\Pr[r_1 \in \mathcal{S}_1] = \sum_{s=W}^{N} \frac{s}{N} \Pr[S(t + Z_{(1)}) = s]$$
$$= \frac{1}{3} \Pr[S(t + Z_{(1)}) = 1] +$$
$$\frac{2}{3} \Pr[S(t + Z_{(1)}) = 2] + \Pr[S(t + Z_{(1)}) = 3], \quad (19)$$

and

$$\Pr[r_1 \in \mathcal{S}_2] = \sum_{s=W}^{N} \frac{s}{N} \Pr[S(t + Z_{(2)}) = s]$$
$$= \frac{1}{3} \Pr[S(t + Z_{(2)}) = 1]$$
$$+ \frac{2}{3} \Pr[S(t + Z_{(2)}) = 2] + \Pr[S(t + Z_{(2)}) = 3]. \quad (20)$$

Therefore, we can express the probability of inconsistency in this case as follows

$$p_t = \frac{6\xi^3 e^{-2\lambda t}}{(\lambda + 2\xi)(\lambda + 3\xi)} \cdot \left( \frac{2\lambda}{(\lambda + 2\xi)(\lambda + 3\xi)} - \frac{(\lambda - \xi)e^{-\lambda t}}{(\lambda + \xi)(2\lambda + 3\xi)} \right).$$

## ACKNOWLEDGMENT

The author would like to thank Viveck Cadambe and Mohammad Fahim for their helpful comments.

## REFERENCES


[1] N. A. Lynch, *Distributed algorithms*. Elsevier, 1996.

[2] H. Attiya, A. Bar-Noy, and D. Dolev, "Sharing memory robustly in message-passing systems," *Journal of the ACM (JACM)*, vol. 42, no. 1, pp. 124–142, 1995.

[3] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: amazon's highly available key-value store," in *ACM SIGOPS operating systems review*, vol. 41, no. 6. ACM, 2007, pp. 205–220.

[4] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 35–40, 2010.

[5] D. Abadi, "Consistency tradeoffs in modern distributed database system design: Cap is only part of the story," *Computer*, vol. 45, no. 2, pp. 37–42, 2012.

[6] W. Vogels, "Eventually consistent," *Communications of the ACM*, vol. 52, no. 1, pp. 40–44, 2009.

[7] D. Malkhi, M. K. Reiter, A. Wool, and R. N. Wright, "Probabilistic quorum systems," *Information and Computation*, vol. 170, no. 2, pp. 184–206, 2001.

[8] X. Wang, S. Yang, S. Wang, X. Niu, and J. Xu, "An application-based adaptive replica consistency for cloud storage," in *2010 Ninth International Conference on Grid and Cloud Computing*. IEEE, 2010, pp. 13–17.

[9] S. Sakr, L. Zhao, H. Wada, and A. Liu, "Clouddb autoadmin: Towards a truly elastic cloud-based data store," in *2011 IEEE International Conference on Web Services*, 2011, pp. 732–733.

[10] H. Wada, A. Fekete, L. Zhao, K. Lee, and A. Liu, "Data consistency properties and the trade-offs in commercial cloud storage: the consumers' perspective." in *CIDR*, vol. 11, 2011, pp. 134–143.

[11] P. Bailis, S. Venkataraman, M. J. Franklin, J. M. Hellerstein, and I. Stoica, "Probabilistically bounded staleness for practical partial quorums," *Proceedings of the VLDB Endowment*, vol. 5, no. 8, pp. 776–787, 2012.

[12] H.-E. Chihoub, S. Ibrahim, G. Antoniu, and M. S. Perez, "Harmony: Towards automated self-adaptive consistency in cloud storage," in *2012 IEEE International Conference on Cluster Computing*, pp. 293–301.

[13] ——, "Consistency in the cloud: When money does matter!" in *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 2013, pp. 352–359.

[14] P. Bailis, S. Venkataraman, M. J. Franklin, J. M. Hellerstein, and I. Stoica, "Quantifying eventual consistency with PBS," *The VLDB Journal*, vol. 23, no. 2, pp. 279–302, 2014.

[15] W. Golab, M. R. Rahman, A. AuYoung, K. Keeton, and I. Gupta, "Client-centric benchmarking of eventual consistency for cloud storage systems," in *2014 IEEE 34th International Conference on Distributed Computing Systems*, 2014, pp. 493–502.

[16] S. Liu, S. Nguyen, J. Ganhotra, M. R. Rahman, I. Gupta, and J. Meseguer, "Quantitative analysis of consistency in nosql key-value stores," in *International Conference on Quantitative Evaluation of Systems*. Springer, 2015, pp. 228–243.

[17] M. McKenzie, H. Fan, and W. Golab, "Fine-tuning the consistency-latency trade-off in quorum-replicated distributed storage systems," in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 1708–1717.

[18] S. Chatterjee and W. Golab, "Brief announcement: A probabilistic performance model and tuning framework for eventually consistent distributed storage systems," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*. ACM, 2017, pp. 259–261.

[19] M. R. Rahman, L. Tseng, S. Nguyen, I. Gupta, and N. Vaidya, "Characterizing and adapting the consistency-latency tradeoff in distributed key-value stores," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 11, no. 4, p. 20, 2017.

[20] J. Zhong, R. D. Yates, and E. Soljanin, "Minimizing content staleness in dynamo-style replicated storage systems," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 361–366.

[21] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, 1987, pp. 1–12.

[22] A. Rényi, "On the theory of order statistics," *Acta Mathematica Hungarica*, vol. 4, no. 3-4, pp. 191–231, 1953.

[23] M. Bibinger, "Notes on the sum and maximum of independent exponentially distributed random variables with different scale parameters," *arXiv preprint arXiv:1307.3945*, 2013.

[24] R. E. Ali, *Harnessing Data Correlation and Network Information in Distributed Key-Value Stores*. PhD Dissertation, Penn State University, 2020.