Enhanced WARS Model Proposal for Advancing Reasoning Consistency Based on Probabilistically Bounded Staleness

Viktor Sobol

School of Mathematics and Computer Science,
V.N. Karazin Kharkiv National University
4, Svobody Sqr., Kharkiv, 61022, Ukraine

Abstract

Systems employing technics of distributing data across multiple machines are widespread nowadays and demand significant expertise from operators. Oftentimes requirement of strong consistency from the data store is too expensive and unaffordable in practical systems. One of the approaches is an application of partial quorum systems with weaker consistency guarantees. Probabilistically Bounded Staleness (PBS) [1] was introduced together with t-visibility. WARS model which is based on mentioned above theory is deliberated to give a tool to reason about datastore consistency by bounding a staleness of data. Further studying of PBS approach is a step towards better understanding and providing tools for reasoning about consistency in distributed systems with partial quorums. The work presented in this paper is a proposition of an enhanced WARS model backed by experimental data to get a more precise view of a system.

Keywords

Adaptive Consistency, Eventual Consistency, Partial Quorums, PBS

1. Introduction

Modern datastore systems are obliged with numerous requirements such as scalability, availability, sufficient level of consistency per an organization's need. Systems operating with significant volumes of data are using different techniques of distributing and replicating data across numerous physical machines satisfy the demands for data quality. With the ability of employing strong consistency guarantees most of the time practitioners are not able to trade off availability [2, 1]. Eventual consistency is used as a consistency model to describe system behaviour, where no guarantees about the staleness of data can be made however, in the absence of new updates the system will result in a coherent state after an undefined time period. Research related to bounding an undefined time frame is ongoing and very important to practitioners. This work is dedicated to study boundaries of eventual consistency in partial quorum systems by using theory presented in [1].

PhD Symposium at ICT in Education, Research, and Industrial Applications co-located with 17th International Conference "ICT in Education, Research, and Industrial Applications 2021" (ICTERI 2021)
viktor.pdt@gmail.com (V. Sobol)
0000-0003-4971-3098 (V. Sobol)

2. Partial quorums and PBS theory

Quorum systems giving a lot of benefits in current datastore architecture and used inside well-known database management systems, such as Cassandra, DynamoDB. The simplified logic of this kind of datastores — is when the write operation arrives at one of the member of the datastore system, the response returned back to the client only after a particular number of members acknowledged this operation, typically called write quorum — W. The same logic applies to read operation with a possible different number of members required to return data, called read quorum — R. In quorum systems, strong consistency is guaranteed when $R + W > N$, where N is the number of replicas storing a data item. Partial quorums, where $R + W \leqslant N$ are used in practice to elevate availability of the system and got proven to be good enough to be widely employed [1]. The work presented in this paper is dedicated to only partial quorum systems. As partial quorums cannot provide strong consistency, numerous research was conducted in order to understand better the level of consistency of such data stores. One of the research is *Probabilistically Bounded Staleness* theory introduced in [1]. Other approaches on this topic are covered in section 6.

PBS $t\text{-}visibility$ models the probability of stale data item being return from the read operation which is initiated after t time units passed since write request is complete. **Definition 3** from [1] is forming an understanding for *t-visibility consistency* and later in this paper revised to include client based view point. *WARS* model was proposed in the same paper as a tool which allow simulation of Dynamo style datastore with a subsequent estimation of *t-visibility*. The model uses probability distributions to model the next steps after the request arrived from a client to one of the nodes of datastore, namely coordinator:

W	distribution of time for coordinator to reach a node storing a replica of data item for a write request;
A	distribution of time for node to send acknowledgment to coordinator;
R	distribution of time for coordinator to reach a node storing a replica of data item for a read request;
S	distribution of node sending a read response back to coordinator;

However, it is complex to study this model analytically, authors of the original paper used Monte Carlo simulation to prove usefulness of a model and theory overall. A similar approach is employed in current work to show the advantages of the proposed incorporation.

3. Enhanced Model proposal

The motivation and impact of the two refinements proposed to the *WARS* model are described in this section below. In the last paragraph of this section, the complete model is expounded.

3.1. Data store request processing time

Cassandra database management system employs query-model first approach as per "Instead of modeling the data first and then writing queries, with Cassandra you model the queries

and let the data be organized around them.", taken from [3]. Hence write and read query performance is highly dependant on the primary design decision. In cases of misreckoning about business requirements or further changing of project necessities which might be supported by new expensive queries for which the original datastore structure was not designed for. As a consequence, the request processing efficiency has to be measured and accounted for the consistency impact it can have. The proposed improvements to the original model can be used to monitor the current state of the datastore and analyze potential change for the positive impact and drawback it can bring.

The result of the provided reasoning above is the inclusion of the write and read request processing time into *WARS* model.

3.2. Client request latency

Authors of the original PBS paper made a straight point that the client delays has to be taken into account while considering practical scenarios [1]. Role of a datastore client can be taken by multiple essences:

- The computational unit inside the same organization software structure. This type of client is usually located in the same network perimeter and delays can be compared by magnitude to the cross node communication inside data storage. However, cross data-center communication despite being in the same network perimeter has a drastic impact on the $t\text{-}visibility$ as shown in the experimentation section.
- The actual client of the organization software. This type of client will incur delays of different magnitude due to browser delays, public network delays, etc. Thus the time between two sequential requests will be much higher.

As per the original definition, $t\text{-}visibility$ is calculated starting from the point after the write request was committed. However, the client is acquainted with the knowledge of the write request being successfully committed only after the response from the coordinator is received. The revised definition of $t\text{-}visibility$ with delineation of client viewpoint is presented below.

Definition 1. *A quorum system obeys PBS $t\text{-}visibility$ consistency from client viewpoint if, with probability $1 - p_{st}$, any client request started at least t units of time after a write response received by a client, returns to a client at least one value that is at least as recent as that write.*

3.3. Enhanced model proposal

The next parameters are included in the original *WARS* model:

D_W	distribution of write request processing time;
D_R	distribution of read request processing time;
C_R	distribution of client request latency;
C_A	distribution of client acknowledgment latency;

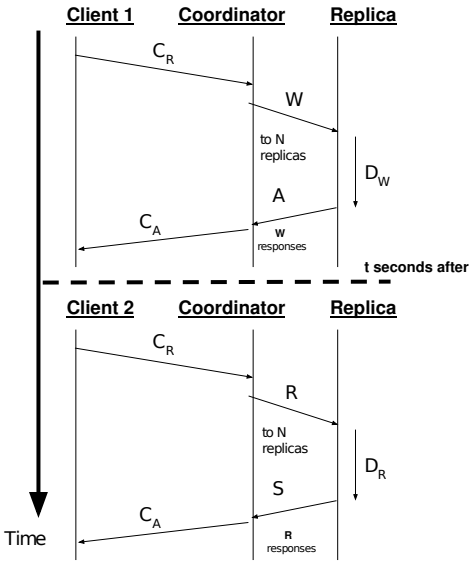

Figure 1: Diagram of *WARS* model with consideration of client-side delays and every replica request processing time.

The entire data flow starting from the client initiating a request can be seen on a space-time diagram shown in Figure 1. The process starts from the client sends a request to the coordinator with the delay drawn from distribution $-C_R$. The coordinator sends N requests to datastore nodes, where N is the amount of replicas. Every request to a datastore instance completes in time composed of value drawn from W, A, and D_W. After the coordinator receives W responses, the response from the coordinator is transmitted back to the client, taking the time of value drawn from C_A. The read part is much alike to the writing part with the difference in the distribution of request delay and datastore processing time values are drawn from. Thus each request out of N from coordinator to instances storing replicas consists of time drawn from R, S, and D_R. The coordinator waits for R responses and then send a response to the client with the latency drawn from C_A

The condition for the client to see a stale data is when read requests from coordinator will be processed faster than write requests on the step before. Deriving from the logic originated in the original paper dedicated to PBS consistency theory [1], the new values are added to the staleness condition. Denoting w_t as the time when the client received a successful response form the coordinator. Then we can consider a stale response from a replica to coordinator in case of $w_t + t + c_R + r' + d_R < w' + d_W$, where $c_R -$ value drawn from C_R, r' from R, w' from W, d_R and d_W from D_R and D_W respectively. The client can receive stale data in case if first R responses to the coordinator are coming from replicas satisfying the condition above. From the replica staleness condition can be seen that tweaking different parameters may improve the chances of getting fresh data. However, accelerating read performance is increasing probability of getting stale data. The analysis using Monte Carlo method together with measurements of $t\text{-}visibility$ of Cassandra datastore in cross data-center environment are presented in the next section.

4. Examination of enhanced *WARS* model

This section will cover a comparison of Monte Carlo simulation of the original *WARS* model and enhanced model with relation to the performance of Cassandra cluster in the multi data-center environment. A synthetic run of the primary model was done according to the simulation described in the original PBS paper in section 5.2 [1].

4.1. Enhanced model Monte Carlo simulation

For the synthetic execution of enhanced model, multiple variations of exponential distribution were taken. The parameters — $\lambda \in \{0.05, 0.1, 0.2, 0.4, 0.5\}$ were used for the network delays from coordinator to replica nodes inside one datastore. For the cases where the data is stored on replicas located in multiple locations exponential distribution with the next parameters was used $-\lambda \in \{0.007, 0.01\}$. For client delays, with consideration of data being stored in multiple locations, it is assumed that distributions for different locations from the client's perspective are not the same. It is considered that one location is closer than the other. First distribution for client delay is $\lambda \in \{0.015\}$ and the second $-\lambda \in \{0.006\}$. The data is taken from measuring network latency to a data center in Frankfurt and Singapore during the actual Cassandra cluster test. The write and read request processing time are considered to be small for simulation described in this paper, hence the distribution are taken with the next parameter for both, read and write request processing time, $\lambda = 1$.

Experiments were conducted with different settings of the amount of replicas — N and read and write consistency — R, W:

- $N = 3,\ R = W = 1$
- $N = 4,\ R = W = 1$
- $N = 4,\ R = W = 2$, cross data-center environment
- $N = 6,\ R = W = 2$, cross data-center environment
- $N = 6,\ R = 2,\ W = 1$, cross data-center environment
- $N = 6,\ R = 1,\ W = 2$, cross data-center environment

The cases for the client to read and write operation from the same data center and different were spread evenly. For the determination of $t\text{-}visibility$, the condition described in section 3.3 is used after the parameters were drawn from the respected distributions.

4.2. Measurements with Cassandra cluster

For measuring $t\text{-}visibility$, Cassandra cluster was prepared using service from cloud vendor — Digital Ocean. All operations were conducted using one process to update a key and multiple processes to read this key from different data centers. Cluster setup consisted of a total of 20 nodes split evenly in Frankfurt and Singapore locations. For every new version update, the difference was measured between events of write request completes and read request obtain the same data.

Density plots of time difference for write operation and then read the corresponding version of data with the respect to geographical locations are shown on Figures 2, 3, 4, 5.

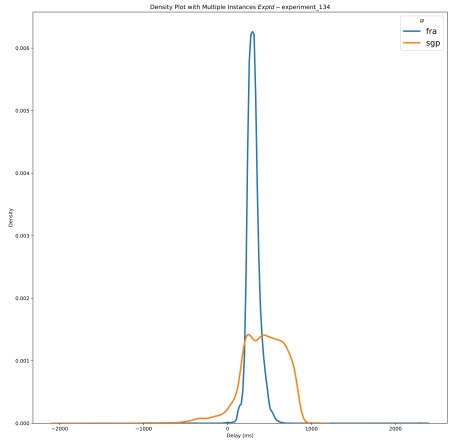

Figure 2: $N = 3,\ W = R = 1$, replicas are located in Frankfurt dc only. Long tails can be observed.

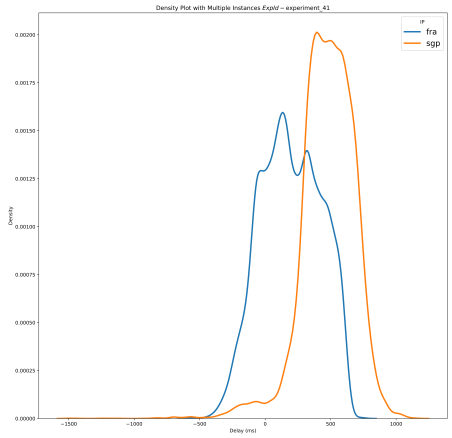

Figure 3: $N = 3,\ W = R = 1$, replicas are located in in both dc. Long tails can be observed.

As can be noticed, cases where the difference in time of mentioned earlier events is negative. The aforementioned can happen due to numerous anomalies in the network such as delays, packet reordering considering a significant variability in distance between client and datastore locations. From $t\text{-}visibility$ prospective negative difference corresponds to state where data is available immediately after the client gets a response from the coordinator. After calculating *RMSE* for Monte Carlo simulation and actual run of the system, the next results were obtained:

- $N = 3,\ R = W = 1$: Enhanced model: 0.5%, Original WARS model: 18.3%
- $N = 4,\ R = W = 1$: 0.7%, Original WARS model: 21%
- $N = 4,\ R = W = 2$, cross data-center environment: Enhanced model: 0.5%, Original WARS model: 19.7%
- $N = 6,\ R = W = 2$, cross data-center environment: Enhanced model: 1.2%, Original WARS model: 18.1%
- $N = 6,\ R = 2,\ W = 1$, cross data-center environment: Enhanced model: 1.7%, Original WARS model: 25.4%

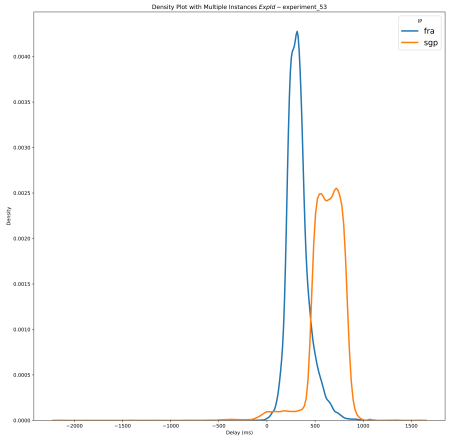

Figure 4: $N = 4,\ W = R = 1$, replicas are located in both dc evenly.

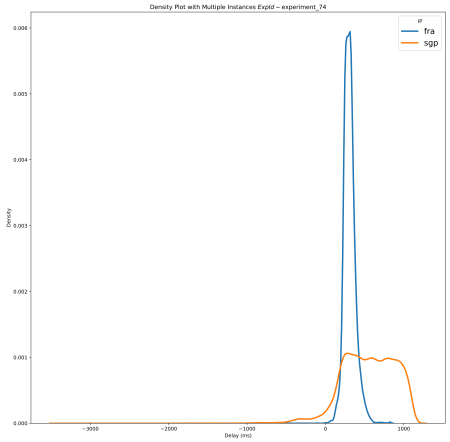

Figure 5: $N = 6,\ W = R = 1$, replicas are located in both dc evenly.

- $N = 6,\ R = 1,\ W = 2$, cross data-center environment: Enhanced model: 1.4%. Original WARS model: 26.2%

The noticeable difference for the original model simulation is explained by not counting client request delays in a geographically distributed environment. The impact of a client delay can be seen on provided graphics. The results are sufficient to conclude that the enhanced model gives a better result for estimation $t\text{-}visibility$ from client side perspective.

5. Related work

This section is dedicated to showing the work based on PBS ideas as well as ideas PBS theory is based on together with alternative approaches in order to achieve the best balance between consistency and availability.

Adaptive consistency model currently gets enough interests and application [4, 5, 6, 7]. PBS

theory is used as a tool for identification of currently possible consistency levels in the system [6]. Harmony framework [8] as alternative to PBS uses *stale read rate* metric dedicated for Cloud storage system, and make an adjustments of the consistency levels based on application needs. As an opposite to adaptive consistency, the Probabilistic Consistency Guarantee approach was studied in [9]. The proposed model is trying to identify the size of a quorum for every read and write request to maximize the chances of reading up to date data along with keeping the throughput steady. TACT algorithm was proposed in [10] by Yu and Vahdat. TACT represents a set of metrics — *Numerical Error, Order Error, Staleness* in order to capture consistency spectrum. Approaches of employing machine learning methods to quantify possible consistency guarantees are studied as part of research [11].

Closed-form expression of PBS *t-visibility* was proposed as an alternative approach to use event-based Monte Carlo simulator in [12]. It is said by the author that the original WARS model simulation was unable to provide precision given by the proposed expression. The direction of comparing the enhanced WARS model to the provided expression together with studying client viewpoint and its relation to the equation is a promising research step.

6. Future work

Optimizing network topology: As an updated model takes into account message reordering and delays in both network connection to datastore and inside datastore instances. The network topology might be optimized per the required need of every organization. A study of the relationship between network topology and consistency was conducted in [13] based on datastore model introduced in [14] and obtained results can be used together with t-visibility approach to improving the balance between performance and consistency of the datastore.

Cost optimization of cloud providers: With the rise of cloud computing services and increasing adoption from the business. Customers of the mentioned services are facing numerous challenges and issues such as vendor locks, capitally studied in [15, 16] and cost optimization challenges [17, 18]. To tackle such formidable tasks organizations are obliged to experiment with different infrastructure setups which are time and resource-consuming actions. The experimentation are required due to the unique needs of every organization. By using an analytical model of the desired system, currently speaking, of the desired datastore, with organization-specific requirements and the SLA from cloud vendors the decision might be less expensive. However, further detailed studying is required in this direction.

Further model development: As request processing time was introduced in an updated version of the model. The model can be developed further by study a dependency between system load and data store request processing time. Resources that can be encountered are the next:

- CPUs: sockets, cores, hardware threads (virtual CPUs)
- Memory: capacity
- Storage devices: I/O, capacity
- Interconnects: CPUs, memory, I/O.

Exploring the influence of the mentioned parameters to the consistency from t-visibility prospective and further embedding into analytical reasoning can be done by using the USE method introduced by Brendan Gregg in [19].

7. Conclusion

During work on this paper, PBS t-consistency analysis from the client perspective was conducted by using data centers located in multiple geographical regions for Cassandra datastore cluster. Definition of $t\text{-}visibility$ with consideration of a client view-point was proposed. The carried experiments allowed us to affirm a sufficiency of the enhanced version of WARS model with a consideration of client delays and data processing time. The new model has been studied by using Monte Carlo simulation and was compared to the measurements procured from the Cassandra cluster execution. The enhanced model gives a more precise picture of data store consistency from client perspective by relying on $t\text{-}visibility$ as a key metric, with consideration of auxiliary parameters. The next actions towards improving and employing the proposed model were defined. Moreover, obtained results can be fitted into the observability and monitoring practices of organization [20] by engaging approaches mentioned in section 5.

Acknowledgments

The author thanks to Prof. Grygoriy Zholtkevych for his supervision and support.

References

[1] B. Peter, V. Shivaram, F. M. J., H. J. M., S. Ion, Probabilistically bounded staleness for practical partial quorums, Proc. VLDB Endow. 5 (2012) 776–787. URL: https://doi.org/10.14778/2212351.2212359. doi:10.14778/2212351.2212359.

[2] D. Abadi, Consistency tradeoffs in modern distributed database system design: Cap is only part of the story, Computer 45 (2012) 37–42. doi:10.1109/MC.2012.33.

[3] J. Carpenter, E. Hewitt, Cassandra: the definitive guide, O'Reilly, 2016. URL: https://www.amazon.de/Cassandra-Definitive-Guide-Jeff-Carpenter/dp/1491933666.

[4] E. Sakic, F. Sardis, J. W. Guck, W. Kellerer, Towards adaptive state consistency in distributed sdn control plane, in: 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–7. doi:10.1109/ICC.2017.7997164.

[5] E. Sakic, W. Kellerer, Impact of adaptive consistency on distributed sdn applications: An empirical study, IEEE J.Sel. A. Commun. 36 (2018) 2702–2715. URL: https://doi.org/10.1109/JSAC.2018.2871309. doi:10.1109/JSAC.2018.2871309.

[6] F. Bannour, S. Souihi, A. Mellouk, Adaptive quorum-inspired sla-aware consistency for distributed sdn controllers, 2019 15th International Conference on Network and Service Management (CNSM) (2019) 1–7.

[7] K. Abdennacer, S. Benharzallah, L. Kahloul, R. Euler, L. Abdelkader, A. Bounceur, A comparative analysis of adaptive consistency approaches in cloud storage, Journal of Parallel and Distributed Computing 129 (2019). doi:10.1016/j.jpdc.2019.03.006.

[8] H.-E. Chihoub, S. Ibrahim, G. Antoniu, M. S. Pérez, Harmony: Towards automated self-adaptive consistency in cloud storage, in: 2012 IEEE International Conference on Cluster Computing, 2012, pp. 293–301. doi:10.1109/CLUSTER.2012.56.

[9] X. Yao, C.-L. Wang, Probabilistic consistency guarantee in partial quorum-based data store, IEEE Transactions on Parallel and Distributed Systems 31 (2020) 1815–1827. doi:10.1109/TPDS.2020.2973619.

[10] H. Yu, A. Vahdat, Design and evaluation of a conit-based continuous consistency model for replicated services, ACM Trans. Comput. Syst. 20 (2002) 239–282. URL: https://doi.org/10.1145/566340.566342. doi:10.1145/566340.566342.

[11] S. Sidhanta, W. Golab, S. Mukhopadhyay, S. Basu, Adaptable sla-aware consistency tuning for quorum-replicated datastores, IEEE Transactions on Big Data 3 (2017) 248–261. doi:10.1109/TBDATA.2017.2656121.

[12] R. Ali, Consistency analysis of replication-based probabilistic key-value stores, ArXiv abs/2002.06098 (2020).

[13] V. Sobol, Simplifying simulation of distributed datastores based on statistical estimating cap-constraint violation, in: Proceedings of the PhD Symposium at ICT in Education, Research, and Industrial Applications co-located with 16th International Conference "ICT in Education, Research, and Industrial Applications 2020 (ICTERI 2020)", volume 2791, CEUR Workshop Proceedings, 2020. URL: http://ceur-ws.org/Vol-2791/2020200042.pdf.

[14] K. Rukkas, G. Zholtkevych, Distributed datastores: Towards probabilistic approach for estimation of reliability., in: Proceedings of the 11th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, volume 1356, CEUR Workshop Proceedings, 2015. URL: http://ceur-ws.org/Vol-1356/paper_51.pdf.

[15] P. S. Justin Lerma, Cloud cost optimization: principles for lasting success, 2020. URL: https://cloud.google.com/blog/topics/cost-management/principles-of-cloud-cost-optimization.

[16] J. Opara-Martins, R. Sahandi, F. Tian, Critical analysis of vendor lock-in and its impact on cloud computing migration: A business perspective, J. Cloud Comput. 5 (2016). URL: https://doi.org/10.1186/s13677-016-0054-z. doi:10.1186/s13677-016-0054-z.

[17] E. Weintraub, Y. Cohen, Cost optimization of cloud computing services in a networked environment, International Journal of Advanced Computer Science and Applications 6 (2015) pp. 148–157. doi:10.14569/IJACSA.2015.060420.

[18] E. Weintraub, Y. Cohen, Optimizing Cloud Computing Costs of Services for Consumers, 2019, pp. 83–96. doi:10.4018/978-1-5225-7766-9.ch007.

[19] B. Gregg, Thinking methodically about performance: The use method addresses shortcomings in other commonly used methodologies., Queue 10 (2012) 40–51. URL: https://doi.org/10.1145/2405116.2413037. doi:10.1145/2405116.2413037.

[20] B. Beyer, C. Jones, J. Petoff, N. R. Murphy, Site Reliability Engineering: How Google Runs Production Systems, 1st ed., O'Reilly Media, Inc., 2016.