UDC 004.942 **doi: 10.20998/2522-9052.2020.2.09**

K. Rukkas, G. Zholtkevych

V. N. Karazin Kharkiv National University, Kharkiv, Ukraine

PROBABILISTIC MODEL FOR ESTIMATION OF CAP-GUARANTEES FOR DISTRIBUTED DATASTORE

Abstract. The **subject** of the article's research is the CAP-guarantees of distributed datastore. The **goal** is to evolve decision-making algorithm for the distributed datastore architecture design which will balance CAP-guarantees depending on business requirements. To achieve that the following problems were solved in the paper: the stochastic model to evaluate different components of CAP-characteristics and some metrics that will impact on these values were developed. To solve these problems the following **methods** were used: basics from graph theory and probability theory, general formulas of expected value and automaton models and software application for calculation of developed formulas. The capability to measure such metrics **resulted in** to forming some constitutes of decision-making algorithm. **Conclusions**: the developed components of decision-making algorithm were the purpose of this paper and it could be one of basic components on the design distributed datastores stage, so that architects who build new software design may also use the algorithm to achieve balanced guarantees of distributed system reliability at the earlier stage of business needs implementation.

Keywords: CAP-guarantees; distributed data warehouses; decision making algorithm; stochastic metrics; average replica propagation time; design of distributed data warehouses.

Introduction

Nowadays all the scalable software needs a reliable storage that responds in a reasonable amount of time and is evolving with every business requirement coming. This comes up with horizontal scaling need, including storage. In the most of cases distributed datastore are applicable for such software. Though scaling distributed datastores often meets the CAP-theorem as a difficulty to overcome. CAP-theorem says that it is impossible to satisfy all three guarantees of a distributed datastore (consistency, availability, partition tolerance). This paper is devoted to finding as generic as possible solutions that will balance these guarantees for specific business needs and will become the base of decision-making algorithm on the stage of build and design of a distributed network and. In the article the mathematical model that presents stochastic formula to measure the probability of delivery in a datastore with imperfect partition tolerance and various kinds of data loss. This will help to evaluate different versions of topology for distributed datastore network and continue the development of the decision-making algorithm for design a datastore with balanced guarantees that corresponds specific business requirements.

Related research

There are many research works devoted to scaling distributed systems, building reliable network and distributed datastores monitoring. So, the estimates of distributed datastores scale is presented in [1]. General research for distributed systems architecture, recommendations of how to build such systems is presented in [2]. Before final proof of CAP-theorem Brewer had investigated the capability of distributed systems to be robust, strong consistency in such system and basically available system with partition tolerance satisfied [3]. But in a few years he presented CAP-theorem as a hypothesis at first and the formal proof of theorem was presented in [4] and some clarifications were made in 2012 [5]. These research works has resulted in the overview and consequences of CAP-theorem [6]. The theorem has been reviewed again in [7].

Since there the CAP-theorem problem got under deep research during the latest 7 years. Building any architecture there is a need to make a choice: ACID or BASE model, strong consistency which results in weaker availability or basically available eventually consistent system where strong consistency is neglected in some point of view. In general a lot of solutions come in BASE model while nowadays in the conditions of network speed growth we cannot sacrifice availability and this model and comparing it with ACID is presented in [8]. The problem of that it is impossible to fulfill all three guarantees of reliable datastore at once is also researched from different points: in the paper of CAP-theorem analysis the circumstances to achieve compromise between these guarantees are considered (see [9]). The work is mostly devoted to the CAP-theorem analysis. But it does not declare precise model to overcome current problems. There are works that present deeper investigation and has developed algorithms for strong consistency balancing (see [10]). Also, the paper [11] showed the general advisory for data replication in distributed systems.

Research aims and objectives. So we can see that the CAP-theorem problem has been investigated from different parts and domains. However, there are no articles devoted to evaluating the CAP-guarantees in stochastic way and there is no general algorithm of design distributed system network, so that CAP-guarantees are satisfied as more as needed for specific business needs. This research is devoted to evolving probabilistic model to measure CAP-guarantees somehow. This will help plenty of systems in monitoring reliability of datastores and invent new algorithm that will contain recommendations on how to design such datastores, so that CAP-guarantees will be fulfilled optimally for any business needs.

Research bases

In the previous section it can be understood that the CAP-theorem problem has been investigated and overcome from different parts. Mostly research is based on performance increase while requirements can grow

much faster. Here we present mathematical model that will allow to evaluate the probability of delivery in the whole datastore. Thus, the model is defined as:

$$(N, L, \partial, D, r, N_d, l(N_d), n_c), \quad (1)$$

where N – finite set of nodes in a datastore; L – finite set of links in a datastore; $\partial : L \to 2^N$ – mapping where each link is associated with two adjacent nodes; D – finite set of stored data units; $r : D \to 2^N$ – mapping that associates each of data unit to a set of nodes that store the replica of this data unit; N_d – finite set of nodes that store the given data unit d; $l(N_d)$ – the number of nodes that store data unit d; n_c – the number of nodes in a subset N_d, where all the nodes have the same version of replica.

Now let us for the next components of such a system. Let we have a graph that represents the topology of distributed network and a subset of paths in a graph P: P_1..., ..., P_n. For now it is necessary to compute the probability of delivery via any path in a graph. Let us assume that one message has to be delivered through path P. Let us calculate the delivery probability in the conditions of data loss on nodes (in the case of partitions etc.) and links. We denote as p_k the probability of delivery through path P. The events of delivery to path nodes are dependent each on other because if a message will not be delivered to j node, it will not be delivered to $j+1$ (де j $\in$ P). Thus, the delivery probability through path P_k:

$$p_k = p_{k1} \cdot \ldots \cdot p_{kn} = \prod_{j=1}^{n} p_{kj}, \quad (2)$$

where $k \in P$, $j \in P_k$. Below we will concerned with two examples. First one is the case when there is no network partitions and the second when partitions do occur. In the first case p_{kj} is the delivery probability through link j. In the second case p_{kj} is the product of delivery probability through link j and the probability of that adjacent node is alive and is able to accept a message. Hence, we can derive the delivery probability through one path P_k, $k \in P$ – the set of all paths from i to j.

$$P_k(dataloss) = 1 - p_k = 1 - \prod_{j=1}^{n} p_{kj}. \quad (3)$$

In a distributed network with a topology graph degree more than 1 there is not the only path a message can be delivered through (see definition of graph degree in [12]). Therefore let us consider the delivery probability through different paths. Basing on epidemic protocols of replicas broadcast applied for many live systems in a distributed network (see gossip, … in [13], [14]) we assume that the message broadcast occurs in parallel through different paths, thus, probabilistic events of delivery are independent. Let us evaluate the probability of data loss for every path $P_k \in P$:

$$P(loss) = \prod_{k=1}^{n} p_k(loss) = \prod_{k=1}^{n} (1 - \prod_{j=1}^{m} p_{kj}), \quad (4)$$

where n is a number of elements in the set P and m is the distance of every path. Taking into account the independence of delivery events via different paths, the delivery probability through at least one path is equal to:

$$P(delivery) = 1 - \prod_{k=1}^{n} p_k(loss) = 1 - (\prod_{i=1}^{n} (1 - \prod_{j=1}^{m} p_{ij})). \quad (5)$$

Hence, now we have the probability of data loss across all paths from i to j nodes and delivery probability through at least one path. Let we have the probabilistic space Ω, that contains two events: A – a message is delivered through P_k, B – a message is delivered at least through one path. The event A might occur only if B occurs. Let us compute So that we calculate the conditional probability with a formula ([15]):

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)}{P(B)} = \frac{p_k}{p(delivery)}, \quad (6)$$

where $k \in P$. Having the delivery probability at least through one path from i to j from the set P, the data loss probability in these conditions is equal to:

$$1 - \frac{p_k}{p(delivery)}. \quad (7)$$

Now we introduce such variable t_k as the time of delivery from i to j that is measured in time slots (of the same measure unit that link cost is measured by in the network. Let every path have own delivery time t_k in the conditions of data loss in a network. Then having formulae of probability, we can derive the meantime of delivery taking into account network data loss. (see formulae of the expected value [16]):

$$\mu_{ij} = \sum_{k=1}^{n} t_k \cdot \left(1 - \frac{p_k}{p(delivery)}\right). \quad (8)$$

So, we have evolved our mathematical model with derived formulas of delivery probability and the mean time delivery of messages in data loss conditions in a distributed datastore. Now we would like to present the practical examples of the theory built above.

To be more intuitive and to get handy experience, we take the graph with 5 nodes and 7 links. Let us firstly introduce the delivery time and delivery probability for every link. We assume that as input data, which can be obtained by network routing metric measurements ([17]), where administrators with a help of monitoring systems can obtain the results of delivery time from node to a neighbor or they can evaluate the probability of delivery from node to neighbors. So we assume these initial metrics are real. So, the initial adjacent matrix of delivery probabilities is the following:

$$P = \begin{pmatrix} 0 & 0.72 & 0.92 & 0 & 0 & 0.74 \\ 0.72 & 0 & 0.94 & 0.73 & 0 & 0 \\ 0.92 & 0.94 & 0 & 0 & 0.91 & 0.86 \\ 0 & 0.73 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.91 & 0 & 0 & 0 \\ 0.74 & 0 & 0.86 & 0 & 0 & 0 \end{pmatrix}, \quad (9)$$

where p_{ij} is the current delivery probability. The elements of matrix where $p_{ij} = 0$ are probabilities of nodes that are adjacent to themselves and those that are not neighbors of current node. This is the auxiliary matrix for obtaining search matrix of mean time delivery. Initial adjacent matrix of time delivery is the following:

$$T = \begin{pmatrix} 0 & 2.5 & 1.2 & 0 & 0 & 1.8 \\ 2.5 & 0 & 1.7 & 1.6 & 0 & 0 \\ 1.2 & 1.7 & 0 & 0 & 2.4 & 2.9 \\ 0 & 1.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.4 & 0 & 0 & 0 \\ 1.8 & 0 & 2.9 & 0 & 0 & 0 \end{pmatrix}, \quad (10)$$

where t_{ij} us the current time delivery for i, j node in a graph. $t_{ij} = 0$ is the time for nodes that are not neighbors and will be calculated with formulas presented above. Thus, the delivery probabilities and delivery time for each of link in a network is obtained with a help of programming model (see DDS Datastore Simulation), that computes the given formula which is the man time of delivery for every path from i to j, we obtained the matrix of mean time delivery. The matrix is symmetric, because the delivery time from i to j should equal to the delivery time from j to i, and, obviously, the delivery time from i to i is equal to 0:

$$\mu = \begin{pmatrix} 0 & 2.5 & 1.2 & 7.18 & 5.81 & 1.8 \\ 2.5 & 0 & 1.7 & 1.6 & 8.23 & 7.24 \\ 1.2 & 1.7 & 0 & 8.69 & 2.4 & 2.9 \\ 7.18 & 1.6 & 8.69 & 0 & 12.47 & 13.25 \\ 5.81 & 8.23 & 2.4 & 12.47 & 0 & 7.24 \\ 1.8 & 7.24 & 1.9 & 13.25 & 7.24 & 0 \end{pmatrix}. \quad (11)$$

But this matrix is calculated with a condition of the perfect partition tolerance in a network, i.e. there are no network partitions. Let us introduce the probability of node aliveness, i.e. the probability of that a node will be able to transfer replica and replace own replica if needed. So taking into account these conditions, we obtain the new matrix of delivery mean time:

$$\mu = \begin{pmatrix} 0 & 2.5 & 1.2 & 9.72 & 8.7 & 1.8 \\ 2.5 & 0 & 1.7 & 1.6 & 11.6 & 12.26 \\ 1.2 & 1.7 & 0 & 10.96 & 2.4 & 2.9 \\ 9.72 & 1.6 & 10.96 & 0 & 15.36 & 17.61 \\ 8.7 & 11.63 & 2.4 & 15.36 & 0 & 10.54 \\ 1.8 & 12.26 & 2.9 & 17.61 & 10.54 & 0 \end{pmatrix}. \quad (12)$$

Evolving decision-making algorithm for distributed network topology design

Based on calculated matrix in the previous section, we are able to construct the following elements of decision-making text algorithm for design stage of network for a distributed datastore.

Algorithm. Input. Initial graph G representing the topology of distributed datastore network. Program T, that calculates the matrix of delivery mean time for each of datastore nodes. Set of delivery optimization algorithms A. Output. Optimal allocation of nodes in the graph in the form of mapping that associates each nodes to its neighbors.

0. Calculate mean time on graph G using program T.

1. Sort nodes in the next order: the first node is the one where read requests come most rarely, and the last one is the node where read requests come most often. Denote this set as N_i.

2. Go to instruction 0, giving program T the input data as initial measurements of delivery probabilities and delivery time from every node to neighbors.

3. Assign roles to nodes from N_i in the next manner: node that have maximal delivery associates with the very first node from N_i and node, that have minimal delivery time, associates with the very last node.

4. Save obtained graph and execute instruction 0.

5. If there are nodes that require fast delivery and there those that do not, swap more important nodes with less ones. Save obtained graph.

6. Execute instruction 0. After that if there are still nodes where mean time do not satisfy required one, use algorithm set to decrease delivery time without decreasing measured value of consistency (consistency in a datastore can be measured using same model from [18]). Otherwise go to 7.

7. If there are nodes that do not require fast delivery, remove not needed links that will increase system cost.

8. Execute instruction 0 for current graph and if mean time delivery decreased for node threshold, go to instruction 7 and repeat it for other nodes until optimal solution will not be found.

Conclusions

This research is devoted to mathematical model for distributed datastore that is evolved with the stochastic formula which allow to measure the mean time delivery in the conditions of data loss in a datastore. The input data for such measurements are based on network metric algorithms that allows to measure initial needed values: the delivery and time probability from node to node. This research results in a formed component of decision-making algorithm on the design stage of a network topology for a datastore^ the technique of nodes allocation depending on implemented mathematical model has been developed. The algorithm will help to find the optimal topology that will allow to achieve as maximum as possible delivery time in the conditions of data loss probability without losing consistency measured value. Then base CAP-guarantees should be balanced due to this work. In the next research current mathematical model will advance in the direction of increasing all the three CAP-guarantees and expansion of algorithm for building the architecture of a distributed datastore general solution.

REFERENCES

1. Kuhlenkamp, J., Klems, M. and Röss, O. (2014), “Benchmarking scalability and elasticity of distributed database systems”, *Proceedings of the VLDB Endowment*, Vol. 7(12), pp. 1219-1230, DOI: https://doi.org/10.14778/2732977.2732995
2. Gupta, S., Saroha K. (2011), “Fundamental Research of Distributed Database”, *IJSMS*, 11.
3. Brewer, E. (2000), “Towards robust distributed systems (abstract)”, *Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing - PODC '00*, DOI: https://doi.org/10.1145/343477.343502
4. Gilbert, S. and Lynch, N. (2002), “Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services”, *ACM SIGACT News*, Vol. 33(2), p. 51, DOI: https://doi.org/10.1145/564585.564601
5. Brewer, E. (2012), “CAP twelve years later: How the "rules" have changed”, *Computer*, Vol. 45(2), pp. 23-29, DOI: https://doi.org/10.1109/MC.2012.37

6. Gilbert, S. and Lynch, N. (2012), "Perspectives on the CAP Theorem", *Computer*, Vol. 45(2), pp. 30-36, DOI: https://doi.org/10.1109/MC.2011.389
7. Kleppmann, M. (2015), "A Critique of the CAP Theorem", *ArXiv*, 1509.05393v2, DOI: https://doi.org/10.17863/CAM.13083
8. Banothu, N., Bhukya, S. and Sharma, K. (2016), "Big-data: Acid versus base for database transactions", *2016 International Conference ICEEOT*, DOI: https://doi.org/10.1109/ICEEOT.2016.7755401
9. Bailis, P. and Ghodsi, A. (2013), "Eventual consistency today: limitations, extensions, and beyond", *Communications of the ACM*, Vol. 56(5), p. 55, DOI: https://doi.org/10.1145/2460276.2462076
10. Calder, B. (2011), "Windows Azure Storage: a highly available cloud storage service with strong consistency", *Proc.of the Twenty-Third ACM SOSP '11*, DOI: https://doi.org/10.1145/2043556.2043571
11. Kumalakov, B. and Bakibayev, T. (2017), "Distributed Data Store Architecture Towards Colonial Data Replication", *2017 IEEE 11th Int. Conf. on Application of Inf. and Comm. Techn. (AICT)*, DOI: https://doi.org/10.1109/ICAICT.2017.8686925
12. Diestel, R. (2016), *Graph theory*, Heidelberg: Springer-Verlag, pp. 5-6, DOI: https://doi.org/10.1109/ICAICT.2017.8686925
13. Burmester, M., Le, T. and Yasinsac, A. (2007), *Adaptive gossip protocols: Managing security and redundancy in dense ad hoc networks*, *Ad Hoc Networks*, Vol. 5(3), pp. 313-323, DOI: https://doi.org/10.1016/j.adhoc.2005.11.007
14. Haas, Z., Halpern, J. and Li Li (2002), "Gossip-based ad hoc routing", *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societiesm*, 3, pp.1707-1716.
15. Gut, A. (2012), *Probability*, Springer, New York, NY: p.17.
16. Ross, S. (2007), *Introduction to probability models*, 9th ed. Amsterdam: Elsevier/Academic Press.
17. Olifer, N. and Olifer, V. (2010), *Computer networks,* Willy India, New Delhi.
18. Rukkas, K. and Zholtkevych, G. (2015), "Distributed Datastores: Towards Probabilistic Approach for Estimation of Dependability", *11th Int. Conference on ICT in Education, Research, and Industrial Applications*, 1356, pp.523-534.

Received (Надійшла) 26.02.2020
Accepted for publication (Прийнята до друку) 22.04.2020

ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

Руккас Кирило Маркович – доктор технічних наук, доцент, професор кафедри теоретичної та прикладної інформатики, Харківський національний університет імені В.Н. Каразіна, Харків, Україна;
Kyrylo Rukkas – Doctor of technical sciences, assistant professor, Karazin Kharkiv National University, Kharkiv, Ukraine;
e-mail: krukkas@gmail.com; ORCID ID: http://orcid.org/0000-0002-7614-0793

Жолткевич Галина Григоріївна – дослідник, інженер – розробник програмного забезпечення, Харків, Україна;
Galyna Zholtkevych – Researcher, Software Engineer, a private entrepreneur, Kharkiv, Ukraine.
e-mail: galynazholtkevych1991@gmail.com; ORCID ID: http://orcid.org/0000-0002-9772-4691

Стохастична модель для оцінки CAP-гарантій для розподілених баз даних

К. М. Руккас, Г. Г. Жолткевич

Анотація. Предметом дослідження статті є CAP-гарантії розподілених баз даних. **Метою** є розвиток алгоритму прийняття рішень для проектування розподілених сховищ даних, який збалансує CAP-гарантії залежно від бізнес потреб. Для досягнення мети були поставлені та вирішені наступні **задачі**: розвинена стохастична модель для оцінки різних компонентів CAP-характеристик та метрик, які впливають на ці значення. Для вирішення задач застосовувались наступні **методи**: базові поняття та визначення з теорії графів та теорії ймовірності, загальні формули математичного сподівання та імітаційні моделі для розподіленого сховища даних, програмне забезпечення, яке використовувалось для підрахунку виведених формул. Технічна можливість вимірювання таких метрик, які сприяють на CAP-характеристики, дало змогу отримати як **результат** формування складових алгоритму прийняття рішень. **Висновки**: розроблені компоненти алгоритму прийняття рішень є метою цієї статті та можуть застосовуватися як базові компоненти на етапі проектування розподілених сховищ даних, отже архітектор програмного забезпечення зможе застосувати такий алгоритм для досягнення збалансованих гарантій надійної розподіленої системи на ранньому етапі формування бізнес потреб та реалізації програмного рішення.

Ключові слова: CAP-гарантії; розподілені сховища даних; алгоритм прийняття рішень; стохастичні метрики; середній час розповсюдження реплік; проектування розподілених сховищ.

Стохастическая модель для оценки CAP-гарантий для распределенных баз данных

К. М. Руккас, Г. Г. Жолткевич

Аннотация. Предметом исследования статьи являются CAP-гарантии распределенных баз данных. **Цель** исследования – развитие алгоритма принятия решений для проектирования распределенных хранилищ данных, который сбалансирует CAP-гарантии в зависимости от бизнес-требований. Для выполнения цели были поставлены и решены следующие **задачи**: развита стохастическая модель для оценки различных компонентов CAP-характеристик и их метрик, которые повлияют на их значения. Для решения задач использовались следующие **методы**: базовые понятия и определения из теории графов и теории вероятности, общие формулы математического ожидания и имитационные модели для распределенного хранилища данных, программное обеспечение, которое использовалось для подсчета выведенных формул. Техническая возможность измерения таких метрик, от которых зависят CAP-характеристики, привело к получению формирования составляющих алгоритма принятия решений как **результата**. **Выводы**: разработанные компоненты алгоритма принятия решений, которые являются целью статьи, могут применяться в качестве базовых компонентов на этапе проектирования распределенных хранилищ данных, следовательно, архитектор программного обеспечения получит возможность достижения сбалансированных гарантий надежной распределенной системы на раннем этапе формирования бизнес требований и реализации программного решения.

Ключевые слова: CAP-гарантии; распределенные хранилища данных; алгоритм принятия решений; стохастические метрики; среднее время распространения реплик; проектирование распределенных хранилищ данных.