Abstract. This paper focuses on the contradiction that follows from Brewer's Conjecture for distributed datastores: the need to deliver qualitative data to the user requires a guarantee of consistency, availability and stability of the system at the same time, but Brewer's Conjecture claims that this is impossible. To overcome this contradiction in the paper it is suggested to estimate statistically violation of these guarantees. To implement this idea the interdependencies between the guarantees and indicators of information quality are considered, different kinds of models specifying the general architecture and behaviour of datastores are described, and, finally, the basic metrics of guarantee ensuring are defined. This consideration allows us to formulate several problems that have both theoretical aspects and engineering applications for the improvement of the technology of distributed data processing.

Keywords: distributed datastore, CAP-theorem, information quality, statistic metrics, simulation
Key Terms: DistributedDataWarehousing, ConcurrentComputation

1 Introduction

Nowadays, any kind of human activity requires an infrastructure to support efficiently the corresponding process of decision making. A modern answer to such a requirement is an information system that is an integrated set of components to collect, store and process data, to deliver information, knowledge, and digital products. Today the development trend of Information and Communication Technology is a wide use of networking technologies. Therefore a typical modern information system is a distributed data processing system with a distributed datastore.

Now it is generally accepted that a distributed datastore should guarantee the following properties: consistency, availability, and partition tolerance (C,A,P). They are discussed in papers [3] and [4], but this discussion is too implicit. These works had been critically reviewed in [5]. This criticism is based on the fact that nobody had so far given explicit and rigorous definitions of these guarantees. Taking into account this remark we accept the following understanding as the origin point of our research:

consistency means that all replicas of any data unit contain the same data at the same time point;

availability means that a datastore eventually returns a response (either a success report or a failure notification) on each request;

and finally, ***partition tolerance*** characterizes the ability of a datastore to continue to operate despite arbitrary message losses or failure of part of the system (sometimes to refer to this ability the more general concept **fault-tolerance** is used).

In the ideal case a distributed datastore should provide these guarantees. But as known Brewer's conjecture [3], being called sometimes CAP theorem, says that it is impossible to maintain simultaneously all three guarantees in

asynchronous or partially synchronous network. Taking into account that synchronization of a distributed datastore decreases essentially system throughput we study consequences of Brewer's conjecture for asynchronous distributed datastores.

A lot of research had been wasted at the consideration of CAP-guarantees. As it is impossible to implement all three of them, we suggest a new approach: to characterise stochastically that each of these requirements is fulfilled or not (see Sec. **??**) and do the best for a consumer - provide him with mechanisms used in different datastores for restoring the validity of the CAP-guarantees.

This supposition leads us to the need to develop a simulation framework for supporting numerical experiments to study these mechanisms.

As we said above, the principal objective for an information system design process is to provide a consumer with qualitative information just in time. Therefore we should understand how information quality (IQ) and the CAP-guarantees are connected. In order to clarify this interconnection we give some model of information quality and determine its indicators that depend on providing the CAP-guarantees in Sec. **??**.

In Sec. **??** we present the conceptual model of a distributed datastore proposed as a background for the framework that should be developed. And finally, in Sec. **??** we give rigorous definitions for stochastic criteria of the CAP-guarantees providing, which is based on the presented conceptual model.

Further to avoid cumbersome formulations we shall say "CAP-properties" instead of "ensuring CAP-guarantees".

Introduction that should say about the importance and purpose of this paper . and subject of the paper

2 Metrics for CAP-guarantees

there is description of each guarantee and what we consider, improve, research in this paper. Impact these metrics have each on other and impact on CAP guarantees and how this is solved here in the paper

3 Delivery probabilities metric

4 Fault Tolerance Metric

5 What metrics has solved for CAP guarantees

6 Conclusion

As said above the leading idea of this paper is to suggest an approach for estimating probabilistic metrics of CAP-properties. To do this we need to determine the basic context of our studying.

It is generally accepted (see [1]) that a distributed datastore is a computer network where information is stored on more than one node, often in a replicated fashion.

Moreover, it is important to mention that a researcher has the possibility to observe datastore events in the sequence according to physical time while each datastore node considers the sequence of events with respect to its local time only. Also we assume that datastores at study meet the eventual consistency requirement [2]. It means that after an isolated read request for any data unit all its replicas eventually have the same value.

Thus, we give the rigorous definitions of CAP-properties in this section. To do this, we describe a distributed datastore using the following mathematical model.

Our model is a tuple (N, L, ∂, D, r), where

N	is a finite set, whose elements correspond to nodes of a distributed datastore;
L	is a finite set, whose elements correspond to links of a distributed datastore;
$\partial : L \to 2^N$	is a mapping that associates each link with two nodes that it connects;
D	is a finite set, whose elements correspond to stored data units;
$r : D \to 2^N$	is a mapping that associates each data unit d with a subset of nodes that store its replica.

Thus, firstly, let us introduce the consistency metric. Taking into account that the consistency is the property when for each data unit its replicas have the same value, we shall consider the probability that all data units at distributed datastore are consistent at a certain time moment. Therefore we define the consistency metric in the following manner.

Definition 1 (Consistency metric). *Let $\sigma_t \in \{0, 1\}$ be the random variable that represents one of two events at time point $t \geq 0$:*

$\sigma_t = 0$	*corresponds to the event "there exists a data unit that is not consistent at the time point t" and*
$\sigma_t = 1$	*corresponds to the event "all data units are consistent at the time point t".*

Then the consistency metric $C(t)$ at time point t is defined by the formula

$$C(t) = \Pr(\sigma_t = 1)\,. \tag{1}$$

The second metric estimates the availability guarantee. We propose to measure the extent of availability as meantime between two events: the request has been received by the datastore and the corresponding response [1] has been sent by the datastore. More, formally:

Definition 2. *Let τ_t be the time interval between a request receiving at the time point t and the corresponding response. Then the mean response time is defined by the formula*

$$T(t) = \mathbf{E}[\,\tau_t\,]\,. \tag{2}$$

[1] This response is either a successful report on request or an error message.

Finally, to estimate the third guarantee, called the partition tolerance, we consider the ability of a datastore to survive network partitions, so that the performance of the datastore does not suffer. This definition is more complicated. Let us consider some time point t. At this instant some nodes are alive, but other ones failed. We denote by N_t^a the set of alive nodes ($N_t^a \subset N$). Similarly, we denote by L_t^a the set of alive links that connect alive nodes.

Definition 3. *We shall say that a data unit $d \in D$ is reachable from a node $n \in N_t^a$ at time point t if there exists a path in the graph (N_t^a, L_t^a) from n to some $n' \in N_t^a \bigcap r(d)$.*

Now we can introduce the metric for the partition tolerance using the previous definition.

Definition 4. *Let $\zeta_t \in \{0, 1\}$ be the random variable that represents one of two events at time point $t \geq 0$:*

$\zeta_t = 0$ *corresponds to the event "there exists a data unit that is not reachable from some alive node at time point t" and*

$\zeta_t = 1$ *corresponds to the event "all data units are reachable from any alive node at time point t".*

Then the partition tolerance metric $P(t)$ at time point t is defined by the formula

$$P(t) = \Pr(\zeta_t = 1)\,. \tag{3}$$

7 Delivery probabilities and consistency metric

8 Fault tolerance metric

9 Conclusion

In this paper we have started studying the problem of the quality for distributed datastores. We have proposed the approach to measure the quality properties of a datastore. Therefore we have described CAP-properties and have built the metric system for CAP-properties estimation. We have described the indicators of the information quality and have found interrelations between the information quality and distributed datastores' properties basing on [9].

In order to have a view of the datastore and be able to work with it we have built its structural and behavioural model and based on this knowledge, we specified probabilistic characteristics for CAP-properties measurement.

Thus, the following steps for the problem set in the paper have been done:

- Formulation of understanding the idea: what are CAP-guarantees for distributed datastores indeed;
- Description of the information quality indicators;
- Investigating the connection between CAP-guarantees and information quality model;
- Building the structural and behavioural models for a distributed datastore;

– Forming "CAP-metrics" as the main idea for studying the quality of distributed datastores.

These steps give us possibilities to study CAP-properties fulfillment using the following background: the fault-tolerance mechanisms for asynchronous systems, concurrent programming, algorithms of data propagation in distributed systems, probably, some issues of internet strategy, Queueing Theory, Mathematical Statistics. Fault-tolerance protocols can be used to invent the algorithms for maintaining the partition tolerance guarantee. Obviously, a researcher should have the good background in Graph Theory, Probability Theory and Concurrent Programming in asynchronous distributed systems. Also, as we need to represent experimental results, it is necessary to imitate the model of a distributed datastore.

Summing up now we might set following problems:

– To simulate the model of a distributed datastore;
– To study different mechanisms to estimate the degree of ensuring each guarantee applying specified metrics;
– To analyse discovered mechanisms and their composition;
– To integrate these mechanisms with simulated asynchronous distributed datastore;
– To estimate theoretically the total time complexity for all provided mechanisms;
– To carry out the experimental estimation.

Acknowledgment. Authors express a deep gratefulness to Grygoriy Zholtkevych and Iryna Zaretska for their help and their criticism.

References

1. Pessach, Ya.: Distributed Storage: Concepts, Algorithms, and Implementations. CreateSpace Independent Publishing Platform (2013)
2. Tanenbaum, A.S., Van Steen, M.: Distributed systems. Principles and Paradigms (2nd Edition). Prentice-Hall, Inc. (2006)
3. Gilbert, S., Lynch, N.: Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. ACM SIGACT News. Vol. 33, No. 2, pp. 51–59 (2002)
4. Brewer, E.: CAP Twelve Years Later: How the "Rules" Have Changed. Computer, IEEE Computer Society. Vol. 45, No. 2 (2012)
5. Burgess, M.: Deconstructing the 'CAP theorem' for CM and DevOps `http://markburgess.org/blog_cap.html` (2012)
6. CRG. Information Quality Survey: Administrator's Guide. Cambridge Research Group, Cambridge, MA (1997)
7. Kovac, R., Lee, Y.W., Pipino, L.L.: Total data quality management: the case of IRI. Conf. on Information Quality (Cambridge, MA), pp. 63–79 (1997)
8. English, L.P.: Information Quality Applied: Best Practices for Improving Business Information, Processes and Systems. Wiley (2009)
9. Kahn, B.K., Strong, D.M., Wang, R.Y.: Information Quality Benchmarks: Product and Service Performance. Comm. ACM. Vol. 45, No. 4 (2002)
10. ISO 8000-1:2011. Data Quality. International Organisation for Standartization (2011).