

Википедия

Теорема CAP

Материал из Википедии — свободной энциклопедии

Теорема CAP (известная также как **теорема Брюера**) — эвристическое утверждение о том, что в любой реализации распределённых вычислений возможно обеспечить не более двух из трёх следующих свойств:

- согласованность данных (англ. *consistency*) — во всех вычислительных узлах в один момент времени данные не противоречат друг другу;
- доступность (англ. *availability*) — любой запрос к распределённой системе завершается корректным откликом, однако без гарантии, что ответы всех узлов системы совпадают;
- устойчивость к разделению (англ. *partition tolerance*) — расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

Акроним *CAP* в наименовании теоремы сформирован из первых букв английских наименований этих трёх свойств.

Принцип был предложен профессором Калифорнийского университета в Беркли Эриком Брюером в июле 2000 года^{[1][2]} и впоследствии получил широкую популярность и признание в среде специалистов по распределённым вычислениям^{[3][4]}. Концепция NoSQL, в рамках которой создаются распределённые нетранзакционные системы управления базами данных, зачастую использует этот принцип в качестве обоснования неизбежности отказа от согласованности данных^{[5][6][7]}. Однако многими учёными^[8] и практиками^[9] теорема CAP критикуется за вольность трактовки и даже недостоверность в том смысле, в котором она распространена в сообществе.

Содержание

Обоснования

Следствия

BASE-архитектура

Примечания

Литература

Обоснования

В 2002 году Сет Джилберт и Нэнси Линч из Массачусетского технологического института подобрали формальные модели асинхронных и синхронных распределённых вычислений, в рамках которых показано выполнение теоремы CAP в условиях отсутствия синхронизации (общих часов) у узлов распределённой системы и принципиальную возможность компромисса в частично синхронных системах^[10]. В этой работе «согласованность» в смысле теоремы CAP соотнесена с выполнением первых двух требований ACID — атомарности и согласованности. В дальнейшем, многие практики ссылались на данную работу как на доказательство теоремы CAP^{[4][11][3]}.

Следствия

С точки зрения теоремы CAP, распределённые системы в зависимости от пары практически поддерживаемых свойств из трёх возможных распадаются на три класса — CA, CP, AP.

В системе класса CA во всех узлах данные согласованы и обеспечена доступность, при этом она жертвует устойчивостью к распаду на секции. Такие системы возможны на основе технологического программного обеспечения, поддерживающего транзакционность в смысле ACID, примерами таких систем могут быть решения на основе кластерных систем управления базами данных или распределённая служба каталогов LDAP^[12].

Система класса CP в каждый момент обеспечивает целостный результат и способна функционировать в условиях распада, но достигает этого в ущерб доступности: может не выдавать отклик на запрос. Устойчивость к распаду на секции требует обеспечения дублирования изменений во всех узлах системы, в связи с этим отмечается практическая целесообразность использования в таких системах распределённых пессимистических блокировок для сохранения целостности^[13].

В системе класса AP не гарантируется целостность, но при этом выполнены условия доступности и устойчивости к распаду на секции. Хотя системы такого рода известны задолго до формулировки принципа CAP (например, распределённые веб-кэши или DNS)^[14], рост популярности решений с этим набором свойств связывается именно с распространением теоремы CAP. Так, большинство NoSQL-систем принципиально не гарантируют целостности данных, и ссылаются на теорему CAP как на мотив такого ограничения^[5]. Задачей при построении AP-систем становится обеспечение некоторого практически целесообразного уровня целостности данных, в этом смысле про AP-системы говорят как о «целостных в конечном итоге» (англ. *eventually consistent*)^[15] или как о «слабо целостных» (англ. *weak consistent*)^[16].

BASE-архитектура

Во второй половине 2000-х годов сформулирован подход к построению распределённых систем, в которых требования целостности и доступности выполнены не в полной мере, названный акронимом **BASE** (от англ. *Basically Available, Soft-state, Eventually consistent* — базовая доступность, неустойчивое состояние, согласованность в конечном счёте), при этом такой подход напрямую противопоставляется ACID^[17]. Под *базовой доступностью* подразумевается такой подход к проектированию приложения, чтобы сбой в некоторых узлах приводил к отказу в обслуживании только для незначительной части сессий при сохранении доступности в большинстве случаев^[18]. *Неустойчивое состояние* подразумевает возможность жертвовать долговременным хранением состояния сессий (таких как промежуточные результаты выборки, информация о навигации, контексте), при этом концентрируясь на фиксации обновлений только критичных операций. *Согласованности в конечном счёте*, трактуемой как возможность противоречивости данных в некоторых случаях, но при обеспечении согласования в практически обозримое время, посвящено значительное количество самостоятельных исследований^{[19][20]}.

Примечания

↑ Показывать компактно

- ↑ Брюер, 2000.
- ↑ Джилберт, Линч, 2002, At PODC 2000, Brewer, in an invited talk, made the following conjecture: it is impossible for a web service to provide the following three guarantees: • Consistency • Availability • Partition-tolerance, p. 55.
- ↑ White Paper Beating the CAP Theorem (<http://www.geniedb.com/wp-content/uploads/2011/04/beating-the-cap-theorem-revised.pdf>) (англ.) (PDF). genedb.com (17 April 2011). Проверено 7 июня 2011. Архивировано (<https://www.webcitation.org/69b2XuVHi>) 1 августа 2012 года.
- ↑ *Browne, Julian* Brewer’s CAP Theorem (<http://www.julianbrowne.com/article/viewer/brewers-cap-theorem>) (англ.) (11 January 2009). Проверено 7 июня 2011. Архивировано (<https://www.webcitation.org/69b2YMWV1>) 1 августа 2012 года.
- ↑ Брюер, 2010, Designers of wide-area systems, in which network partitions are considered inevitable, know they cannot have both availability and consistency, and thus can now justify weaker consistency. The rise of the “NoSQL” movement (“Not Only SQL”) is an expression of this freedom, p. 335.
- ↑ Рис, 2011, p. 49.

7. Кузнецов, 2011, Более серьёзным “теоретическим” основанием NoSQL-разработок, в которых общепринятые полезные свойства систем управления данными приносятся в жертву доступности этих систем, является так называемая “теорема CAP”, впервые сформулированная Эриком Брювером, с. 191.
8. Кузнецов, 2011, я заключаю слово теорема в кавычки, поскольку утверждение, названное Брювером теоремой, таковым я признать не могу из-за отсутствия какой-либо чёткой и хотя бы немного формальной постановки задачи ... но широко распространено мнение, что она означает невозможность поддержки в одной распределённой системе управления данными свойств согласованности данных (Consistency), доступности (Availability) и устойчивости к разделению сети (Partitioning), с. 191-192.
9. Рис, 2011, So why are many of the leading social networking sites (Facebook, MySpace, Twitter), e-commerce Web sites (hotel reservation systems and shopping sites), and large banking applications still implemented using traditional database systems such as MySQL (Facebook, Twitter) or SQL Server (MySpace, Choice Hotels International, Bank Itau) instead of using the new NoSQL systems? ... The high-level answer is that the application architecture is still weighing the same trade-offs required by the CAP theorem, p. 50.
10. Джилберт, Линч, 2002, In an asynchronous model, when no clocks are available, the impossibility result is fairly strong: it is impossible to provide consistent data, even allowing stale data to be returned when messages are lost. However in partially synchronous models it is possible to achieve a practical compromise between consistency and availability, p. 59.
11. Григорик, 2010, Problem is, the CAP theorem proposed by Eric Brewer and later proved by Seth Gilbert and Nancy Lynch, shows that together, these three requirements are impossible to achieve at the same time.
12. Картер, 2011, Some CA systems include: single site databases, cluster databases and LDAP.
13. Демидов, 2010, CP (отказы обслуживания) – это подход с наличием дублирования, строгой ACID непротиворечивостью и синхронной репликацией изменений с применением метода пессимистических блокировок.
14. Картер, 2011, Some AP Systems include: Web caching systems and the Domain Name System (DNS).
15. Картер, 2011, Eventual consistency – performing a read operation may give stale data to the client, but all writes will eventually propagate through the entire system.
16. Григорик, 2010, CAP Implies Weak Consistency.
17. Притчетт, 2008, If ACID provides the consistency choice for partitioned databases, then how do you achieve availability instead? One answer is BASE (basically available, soft state, eventually consistent). BASE is diametrically opposed to ACID.
18. Притчетт, 2008, The availability of BASE is achieved through supporting partial failures without total system failure. Here is a simple example: if users are partitioned across five database servers, BASE design encourages crafting operations in such a way that a user database failure impacts only the 20 percent of the users on that particular host.
19. Стоунбрейкер, 2010.
20. Фогельс, 2008.

Литература

- *Brewer, Eric A.* Towards robust distributed systems (<http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-key-note.pdf>) (англ.) // Proceedings of the XIX annual ACM symposium on Principles of distributed computing. — Portland, OR: ACM, 2000. — Vol. 19, no. 7. — DOI:10.1145/343477.343502 (<https://dx.doi.org/10.1145%2F343477.343502>).
- *Brewer, Eric A.* A Certain Freedom: Thoughts on the CAP Theorem (http://portal.acm.org/ft_gateway.cfm?id=1835701&type=pdf&CFID=25475815) (англ.) // Proceeding of the XXIX ACM SIGACT-SIGOPS symposium on Principles of distributed computing. — N.Y.: ACM, 2010. — Iss. 29. — No. 1. — P. 335—336. — ISBN 978-1-60558-888-9. — DOI:10.1145/1835698.1835701 (<https://dx.doi.org/10.1145%2F1835698.1835701>).
- *Carter, Andrew.* The CAP Theorem as it Applies to Contemporary NoSQL Storage Systems (http://www.andrewjonascarter.com/files/ResearchPaper_Databases.pdf) (англ.). Memorial University (22 May 2011). Проверено 11 июня 2011.
- *Демидов А.А.* Проектирование распределённых систем обработки объектных структур данных (<http://rcdl.ru/doc/2010/441-447.pdf>) (рус.) // Труды XII конференции RCDL. — Казань: Казанский государственный университет, 2010. — Вып. 12. — С. 441—447.
- *Gilbert, Seth and Lynch, Nancy.* Brewer’s Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services (<http://lpd.epfl.ch/sgilbert/pubs/BrewersConjecture-SigAct.pdf>) (англ.) // ACM SIGACT News. — ACM, 2002. — Vol. 33, iss. 2. — P. 51—59. — ISSN 0163-5700 (<https://www.worldcat.org/search?fq=x0:jrnl&q=n2:0163-5700>). — DOI:10.1145/564585.564601 (<https://dx.doi.org/10.1145%2F564585.564601>).
- *Grigorik, Ilya* Weak Consistency and CAP Implications (<http://www.igvita.com/2010/06/24/weak-consistency-and-cap-implications/>) (англ.). Igvita (24 June 2010). Проверено 11 июня 2011. Архивировано (<https://www.webcitation.org/67eAqj4W2>) 14 мая 2012 года.
- *Кузнецов, Сергей.* MapReduce: внутри, снаружи или сбоку от параллельных СУБД? (http://www.ispras.ru/ru/proceedings/docs/2010/19/isp_19_2010_35.pdf) (рус.) // Труды Института системного программирования

- РАН. — М.: Институт системного программирования РАН, 2010. — Т. 19. — С. 35—40. — ISSN 2079-8156 (<https://www.worldcat.org/search?fq=x0:jrnl&q=n2:2079-8156>).
- Кузнецов, Сергей. Транзакционные параллельные СУБД: новая волна (http://www.ispras.ru/ru/proceedings/docs/2011/20/isp_20_2011_189.pdf) (рус.) // Труды Института системного программирования РАН. — М.: Институт системного программирования РАН, 2011. — Т. 20. — С. 189—251. — ISSN 2079-8156 (<https://www.worldcat.org/search?fq=x0:jrnl&q=n2:2079-8156>).
 - Pritchett, Dan. BASE: An Acid Alternative (<http://queue.acm.org/detail.cfm?id=1394128>) (англ.) // ACM Queue. — N. Y.: ACM, 2008. — Vol. 6, no. 3. — P. 48—55. — ISSN 1542-7730 (<https://www.worldcat.org/search?fq=x0:jrnl&q=n2:1542-7730>). — DOI:10.1145/1394127.1394128 (<https://dx.doi.org/10.1145%2F1394127.1394128>).
 - Rys, Michael. Scalable SQL. How do large-scale sites and applications remain SQL-based? (<http://cacm.acm.org/magazines/2011/6/108663-scalable-sql/fulltext>) (англ.) // Communications of the ACM. — ACM, 2011. — Vol. 54, no. 6. — P. 48—53. — ISSN 0001-0782 (<https://www.worldcat.org/search?fq=x0:jrnl&q=n2:0001-0782>). — DOI:10.1145/1953122.1953141 (<https://dx.doi.org/10.1145%2F1953122.1953141>).
 - Стоунбрейкер, Майкл. Ошибки в системах баз данных, согласованность "в конечном счете" и теорема CAP (<http://citforum.ru/gazeta/154/>) (рус.). Citforum (19 мая 2010). Проверено 4 июня 2011.
 - Stonebraker, Michael and Cattell, Rick. Scalable SQL. How do large-scale sites and applications remain SQL-based? (<http://cacm.acm.org/magazines/2011/6/108663-scalable-sql/fulltext>) (англ.) // Communications of the ACM. — ACM, 2011. — Vol. 54, no. 6. — P. 72—80. — ISSN 0001-0782 (<https://www.worldcat.org/search?fq=x0:jrnl&q=n2:0001-0782>). — DOI:10.1145/1953122.1953144 (<https://dx.doi.org/10.1145%2F1953122.1953144>).
 - Vogels, Werner. Eventually Consistent (<http://portal.acm.org/citation.cfm?id=1466448>) (англ.) // Queue. — ACM, 2008. — Vol. 6, no. 6. — P. 15—19. — ISSN 1542-7730 (<https://www.worldcat.org/search?fq=x0:jrnl&q=n2:1542-7730>). — DOI:10.1145/1466443.1466448 (<https://dx.doi.org/10.1145%2F1466443.1466448>).

Источник — https://ru.wikipedia.org/w/index.php?title=Теорема_CAP&oldid=89438527

Эта страница последний раз была отредактирована 4 декабря 2017 в 05:15.

Текст доступен по лицензии [Creative Commons Attribution-ShareAlike](#); в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации [Wikimedia Foundation, Inc.](#)

[Свяжитесь с нами](#)