The ACE Theorem for Querying the Web of Data

Jürgen Umbrich
Fujitsu (Ireland) Limited,
Dublin, Ireland

Claudio Gutierrez
DCC, Universidad de Chile,
Santiago, Chile

Aidan Hogan
DERI Galway, NUI Galway,
Ireland

Marcel Karnstedt
DERI Galway, NUI Galway,
Ireland

Josiane Xavier Parreira
DERI Galway, NUI Galway,
Ireland

ABSTRACT

Inspired by the CAP theorem, we identify three desirable properties when querying the Web of Data: Alignment (results up-to-date with sources), Coverage (results covering available remote sources), and Efficiency (bounded resources). In this short paper, we show that no system querying the Web can meet all three "ACE" properties, but instead must make practical trade-offs that we outline.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Design; Theory; Performance

Keywords

Web of Data; Query Processing; ACE Properties

1. INTRODUCTION

Querying the Web of Data gives rise to intuitive trade-offs. Query answering over an index of replicated Web content offers low response times, but query results might be stale due to the dynamic nature of the Web. Alternatives to this method execute queries directly over the Web, delivering fresher results compared to replication, but at the cost of slow accesses to remote sources at runtime. Such trade-offs appear *fundamental* but have not yet been stated in a satisfactory (e.g., formal) manner.

In trying to understand the fundamental limits of what is possible for querying the Web's content—itself an unbounded distributed system—we are inspired by works on distributed computing. In 1994, Peter Deutsch proposed "The Eight Fallacies of Distributed Computing", later extended by James Gosling [2]. Analogously, we previously postulated eight fallacies (assumptions that do not hold in general) specific for querying the Web of Data [3]. We now elaborate on one such fallacy—*"One system can ACE them all"*—which follows the precedent of the CAP theorem for distributed systems [1] and proposes that no system querying the Web can meet the following three expectations:

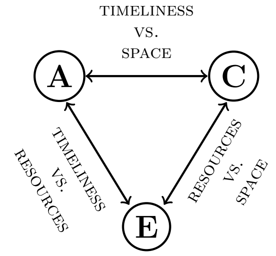


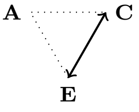
Figure 1: Trade-offs in the ACE triangle

Alignment: The answers given by a query engine are correct and synchronised with respect to sources on the Web at the point when the query is issued.

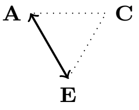
Coverage: The answers given by a query engine are with respect to all sources available on the Web.

Efficiency: The query engine runs with bounded resources, encapsulating various dimensions such as machines, time, energy, parallelism, messages, etc.

In summary, our conjecture is that, using only bounded resources, a query engine cannot guarantee to return all possible results that are synchronised with respect to current content on the unbounded and dynamic Web. Instead, query-engine designers must make practical trade-offs between the ACE properties as depicted in Figure 1. We emphasise that unlike the *discrete* (and achievable) CAP guarantees, each property of ACE is *continuous*, and once a property is fixed at a certain point, the remaining two properties can be considered as antagonistic to each other, as follows.



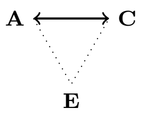
By fixing a certain expectation of alignment, i.e., how timely results must be, a query engine must decide between meeting that expectation efficiently for few sources, or covering more sources at greater cost. This design trade-off is faced by many time-critical or event-based applications, including real-time route planning (e.g., monitor Web sources for recent accidents, traffic jams, etc.) or continuous stock market analytics (e.g., find stock prices, mergers, quarterly reports, etc.) and so forth.



By fixing the sources covered, the degree of alignment will be a function of the cost of keeping local memory (i.e., replicated indexes, soft caches) synchronised with remote Web sources. This design trade-off is faced

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

by vertical search engines that only cover a bounded set of sources, such as flight search mediators (e.g., how often to update flight costs and routes from a fixed set of airlines covered), weather aggregators (e.g., how often to refresh temperatures from available stations), and so forth.



By fixing the amount of resources, such as processing and response time, a query engine must decide between covering many sources but returning less timely results, or covering fewer sources but returning more timely results. This design trade-off is faced by search engines with fixed query-time expectations (e.g., Web search engines which guarantee sub-second response times to millions of users) or resource-constrained devices (e.g., news-feed aggregators on mobile phones, etc.).

These trade-offs together form the cyclical ACE triangle. The system designer must thus compromise in the selection of how many sources to cover, how timely results should be, and how many resources to dedicate. We now provide a formal statement for these intuitions, covering a "discrete" bounded/unbounded model; which abstracts away many (important) parameters like the topology of the Web, the continuous dynamics of sources, variable communication costs, etc.; but nevertheless captures the core of our idea.

2. THE ACE THEOREM

The ACE Theorem builds on the formal modelling of each of the three desirable properties and aims to highlight their antagonistic nature, as well as demonstrating our hypothesis that "one system cannot ACE them all" given a discrete, idealised version of the properties.

Basic ACE Definitions (Discrete version).

Let U be the set of URIs covered. For each $u \in U$, let $D(u)$ be the content obtained (e.g., a set of data items) from accessing URI u. If $D(u) \neq \emptyset$, we call $D(u)$ a source. A source $D(u)$ is dynamic with fixed change rate r_u.

Alignment model: Alignment refers to synchronising with changes in source $D(u)$. This also includes the "creation" of sources where $D(u) = \emptyset$ changes to $D(u) \neq \emptyset$.

Alignment constraint: For t (global clock) and $u \in U$: have $D_t(u)$ in M (local memory).

Alignment working hypothesis: $r_u = 0$ for all $u \in U$.

Coverage model: Coverage refers to the set of URIs u included. The (full) set U is infinite and recursively enumerable. Each $D(u)$ is created distributively. For each time-point t, $D_t(u) \neq \emptyset$ only for finitely many u.

Coverage constraint: At a given time t, have in M all $D_t(u)$ where $D_t(u) \neq \emptyset$.

Coverage working hypothesis: U is finite.

Efficiency model: Efficiency refers to the amount of resources we use. A unit of resource is one access/dereferencing of a URI $u \in U$. We have unbounded memory and local processing capabilities. Cost is the number of parallel accesses possible to URIs in U.

Efficiency constraint: Cost is bounded at each time t. *Moreover*, the function $E(t)$ (amount of resources available) is arbitrary but fixed (i.e., we can define how to augment the amount of resources as time passes, but cannot change the rate of resource growth on demand).

Efficiency working hypothesis: Unbounded resources.

ACE means to have alignment, coverage and efficiency constraints together. This means to ensure, with a bounded number of resources, to have, for any time t, for all $u \in U$ with nonempty $D_t(u)$, all such $D_t(u)$ in M.

Theorem (ACE).

1. Alignment, Coverage and Efficiency constraints cannot be enforced together (even for an *omniscient* agent, i.e., one that instantly knows all possible states of the world).

2. Every pair of constraints in $\{A, C, E\}$ can be enforced assuming the working hypothesis for the third.

Proof (sketch).

1.A. Fix t_0. If A is not an omniscient agent, it will not know which $D_{t_0}(u)$ are non-empty at t_0. Thus coverage cannot be enforced with finite resources.

1.B. Assume that A is omniscient. Denote $S_t = \{u : D_t(u) \neq \emptyset\}$. For a fixed t_0 it can be done: once S_{t_0} is known, dereference it in parallel at t_0. But in general it is not possible: simply consider that the rate of increase of S_t is greater than that of $E(t)$.

2.a. EC. *Hypothesis for Alignment*: $r_u = 0$ for all u. Time is not a constraint anymore. With the bounded amount of resources, dereference all u, one after the other.

2.b. AC. *Hypothesis for Efficiency*: unbounded number of resources. For each t, get in parallel all non-empty $D_t(u)$'s.

2.c. EA. *Hypothesis for Coverage*: U is finite. For each t, get in parallel all non-empty $D_t(u)$'s.

3. CONCLUSION

The three working hypotheses defined above cannot hold for the Web in general, meaning that systems querying in such an environment cannot meet the ideal and "ACE them all". Instead, the ACE theorem shows that the three core properties are functions of each other: E(A,C), A(E,C) and C(A,E). That is, given fixed values of say A and C, one can state the amount of resources E that are necessary; and so on. Understanding the detail of these functions requires a more detailed model and is subject to future work. However, the discrete model we present herein already shows that these trade-offs are *fundamental* and thus should be explicitly considered by designers of Web query engines.

Acknowledgments. This paper was part funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2), Project Fondecyt 1110287 & the European Commission with contract number FP720117287661 (GAMBAS).

4. REFERENCES

[1] E. Brewer. Towards robust distributed systems. In *Principles of Distributed Computing*, July 2000.

[2] A. Rotem-Gal-Oz. Fallacies of distributed computing explained. Technical report, 2012.

[3] J. Umbrich, C. Gutierrez, A. Hogan, M. Karnstedt, and J. X. Parreira. Eight Fallacies when Querying the Web of Data. In *DESWEB Workshop (at ICDE)*, 2013.