

## Storage

Vol. 11 No. 3 – March 2013



### Discrimination in Online Ad Delivery

#### **Google ads, black names and white names, racial discrimination, and click advertising**

Do online ads suggestive of arrest records appear more often with searches of black-sounding names than white-sounding names? What is a black-sounding name or white-sounding name, anyway? How many more times would an ad have to appear adversely affecting one racial group for it to be considered discrimination? Is online activity so ubiquitous that computer scientists have to think about societal consequences such as structural racism in technology design? If so, how is this technology to be built? Let's take a scientific dive into online ad delivery to find answers.

by Latanya Sweeney

### Eventual Consistency Today: Limitations, Extensions, and Beyond

#### **How can applications be built on eventually consistent infrastructure given no guarantee of safety?**

In a July 2000 conference keynote, Eric Brewer, now VP of engineering at Google and a professor at the University of California, Berkeley, publicly postulated the CAP (consistency, availability, and partition tolerance) theorem, which would change the landscape of how distributed storage systems were architected. Brewer's conjecture--based on his experiences building infrastructure for some of the first Internet search engines at Inktomi--states that distributed systems requiring always-on, highly available operation cannot guarantee the illusion of coherent, consistent single-system operation in the presence of network partitions, which cut communication between active servers. Brewer's conjecture proved prescient: in the following decade, with the continued rise of large-scale Internet services, distributed-system architects frequently dropped "strong" guarantees in favor of weaker models--the most notable being eventual consistency.

by Peter Bailis, Ali Ghodsi

### A File System All Its Own

#### **Flash memory has come a long way. Now it's time for software to catch up.**

In the past five years, flash memory has progressed from a promising accelerator, whose place in the data center was still uncertain, to an established enterprise component for storing performance-critical data. Its rise to prominence followed its proliferation in the consumer world and the volume economics that followed (see figure 1). With SSDs (solid-state devices), flash arrived in a form optimized for compatibility - just replace a hard drive with an SSD for radically better performance. But the properties of the NAND flash memory used by SSDs differ significantly from those of the magnetic media in the hard drives they often displace. While SSDs have become more pervasive in a variety of uses, the industry has only just started to design storage systems that embrace the nuances of flash memory. As it escapes the confines of compatibility, significant improvements in performance, reliability, and cost are possible.

by Adam H. Leventhal



## Discrimination in Online Ad Delivery

### Google ads, black names and white names, racial discrimination, and click advertising

Latanya Sweeney

Do online ads suggestive of arrest records appear more often with searches of black-sounding names than white-sounding names? What is a black-sounding name or white-sounding name, anyway? How many more times would an ad have to appear adversely affecting one racial group for it to be considered discrimination? Is online activity so ubiquitous that computer scientists have to think about societal consequences such as structural racism in technology design? If so, how is this technology to be built? Let's take a scientific dive into online ad delivery to find answers.

"Have you ever been arrested?" Imagine this question appearing whenever someone enters your name in a search engine. Perhaps you are in competition for an award, a scholarship, an appointment, a promotion, or a new job, or maybe you are in a position of trust, such as a professor, a physician, a banker, a judge, a manager, or a volunteer. Perhaps you are completing a rental application, selling goods, applying for a loan, joining a social club, making new friends, dating, or engaged in any one of hundreds of circumstances for which someone wants to learn more about you online. Appearing alongside your list of accomplishments is an advertisement implying you may have a criminal record, whether you actually have one or not. Worse, the ads may not appear for your competitors.

Job applications frequently include questions such as: Have you ever been arrested? Have you ever been charged with a crime? Other than a traffic ticket, have you been convicted of a crime? Employers ask these questions to establish trustworthiness. Because others often equate a criminal record with not being reliable or honest, protections exist for those having criminal records.

If an employer disqualifies a job applicant based solely upon information indicating an arrest record, the company may face legal consequences. The U.S. EEOC (Equal Employment Opportunity Commission) is the federal agency charged with enforcing Title VII of the Civil Rights Act of 1964, a law that applies to most employers, prohibiting employment discrimination based on race, color, religion, sex, or national origin. Guidance issued in 1973 extended protections to people with criminal records.<sup>5,11</sup> Title VII does not prohibit employers from obtaining criminal background information. Certain uses of criminal information, however, such as a blanket policy or practice of excluding applicants or disqualifying employees based solely upon information indicating an arrest record, can result in a charge of discrimination.

To make a determination, the EEOC uses an adverse impact test that measures whether certain practices, intentional or not, have a disproportionate effect on a group of people whose defining characteristics are covered by Title VII. To decide, you calculate the percentage of people affected in each group and then divide the smaller value by the larger to get the ratio and compare the result to 80. For example, suppose a company laid off comparable black and white workers at the same rate—25 percent of blacks and 25 percent of whites—then the ratio, 25 divided by 25, would be 100 percent. If the ratio is less than 80 percent, then the EEOC considers the effect disproportionate and may hold the employer responsible for discrimination.<sup>6</sup>

What about online ads suggesting someone with your name has an arrest record, even when no one with your name has been arrested? Title VII does not apply unless you have an arrest record and can prove the potential employer routinely uses ads or information from the company sponsoring the ads, and the result has an inappropriate chilling effect on hiring applicants with criminal records.

The advertiser may argue the ads are commercial free speech—a constitutional right to display the ad associated with your name. The First Amendment of the U.S. Constitution protects advertising. In a landmark decision, the U.S. Supreme Court set out a test for assessing government restrictions on commercial speech, which begins by determining whether the speech is misleading.<sup>3</sup> Are online ads suggesting the existence of an arrest record misleading if no one by that name has an arrest record?

Assume the ads are free speech: what happens when these ads appear more often for one racial group than another? Not everyone is being equally affected by the free speech. Is that free speech or racial discrimination?

*Racism*, as defined by the U.S. Commission on Civil Rights, is “any attitude, action, or institutional structure which subordinates a person or group because of their color . . . Racism is not just a matter of attitudes; actions and institutional structures can also be a form of racism.”<sup>16</sup> *Racial discrimination* results when a person or group of people is treated differently based on their racial origins, according to the Panel on Methods for Assessing Discrimination of the National Research Council.<sup>12</sup> Power is a necessary precondition, because discrimination depends on the ability to give or withhold benefits, facilities, services, opportunities, etc., from someone who should be entitled to them and is denied on the basis of race. *Institutional* or *structural racism*, as defined in *The Social Work Dictionary*, is a system of procedures/patterns whose effect is to foster discriminatory outcomes or give preferences to members of one group over another.<sup>1</sup>

Racism can result, even if not intentional, and online activity now may be so ubiquitous that computer scientists have to think about societal consequences such as structural racism in the technology they design. These considerations frame the big picture, the relevant legal, societal, and technical landscape in which this exploration resides. Now we turn to the exploration itself: whether online ads suggestive of arrest records appear more often for one racial group than another among a sample of racially associated names. Then, we examine the role technology might play in combating this problem if evidence of the pattern exists.

## THE PATTERN

What is the suspected pattern of ad delivery? Here is an overview of the issue with some real-world examples.


This study begins with the assumption that personalized ads suggestive of arrest records do not differ by race. We did this by carefully constructing the scientifically best instance of the pattern—one with names shown to be racially identifying and pseudo-randomly selected.

Earlier this year, a Google search for *Latanya Farrell*, *Latanya Sweeney*, and *Latanya Lockett* yielded the ads and criminal reports shown in figure 1. The ads appeared on Google.com (figure 1a,1c,1e) and on a news Web site, Reuters.com, to which Google supplies ads (figure 1c, bottom). All the ads in question linked to *instantcheckmate.com* (figure 1b,1d,1f). The first ad implied Latanya Farrell may have been arrested. Was she? Clicking on the link and paying the requisite subscription fee revealed

## FIGURE 1

## Sample Ads and Criminal Reports


A

Ads related to latanya farrell 

[Latanya Farrell, Arrested?](http://www.instantcheckmate.com/)  
www.instantcheckmate.com/  
1) Enter Name and State. 2) Access Full Background Checks Instantly.

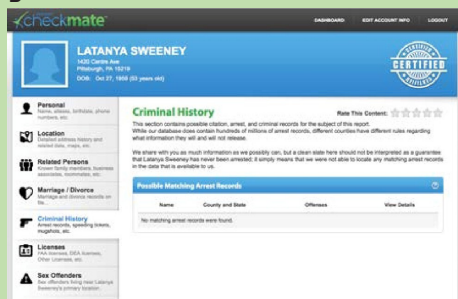
[Latanya Farrell](http://www.publicrecords.com/)  
www.publicrecords.com/  
Public Records Found For: **Latanya Farrell**. View Now.

C


Ad related to latanya sweeney 

[Latanya Sweeney Truth](http://www.instantcheckmate.com/)  
www.instantcheckmate.com/  
Looking for **Latanya Sweeney**? Check **Latanya Sweeney's** Arrests.

D

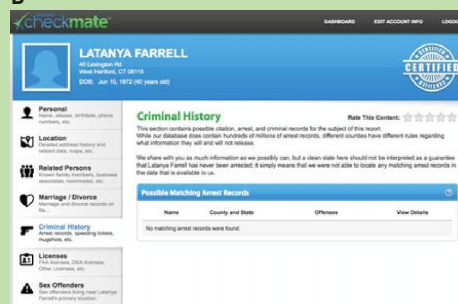


F



Name	County and State	Offenses	View Details
1 Latanya Loretta Lockett	CA Contra Costa County Court	CriminalOff	<a href="#">View Details</a>
2 Latanya Loretta Lockett	CA Contra Costa County	CriminalOff	<a href="#">View Details</a>
3 Latanya M Lockett	CA Orange Superior Court	CriminalOff	<a href="#">View Details</a>
4 Latanya Loretta Lockett	CA Orange Superior Court	CriminalOff	<a href="#">View Details</a>

B



C

Ads by Google

[Latanya Sweeney, Arrested?](http://www.instantcheckmate.com/)  
1) Enter Name and State. 2) Access Full Background Checks Instantly.  
www.instantcheckmate.com/  
[Latanya Sweeney](http://www.publicrecords.com/)  
Public Records Found For: **Latanya Sweeney**. View Now.  
www.publicrecords.com/  
[La Tanya](http://www.ask.com/La+Tanya)  
Search for La Tanya Look Up Fast Results now!  
www.ask.com/La+Tanya

E

Ads related to latanya lockett 

[We Found:Tanya Lockett](http://www.peoplesmart.com/)  
www.peoplesmart.com/  
1) Get Tanya **Lockett's** Info - Try Free! 2) Current Phone, Address & More.

[Latanya Lockett, Arrested?](http://www.instantcheckmate.com/)  
www.instantcheckmate.com/  
1) Enter Name and State. 2) Access Full Background Checks Instantly.

[Latanya Lockett,Found](http://www.whitepages.com/Latanya+Lockett)  
www.whitepages.com/Latanya+Lockett  
Don't Pay for Info that's Free, Get Address, Phone, Photos, & More!  
Name Popularity & Facts - Neighbor Search - Reverse Phone Lookup

that the company had no arrest record for her (figure 1b). There is no arrest record for *Latanya Sweeney* either, but there is for *Latanya Lockett*.

In comparison, searches for *Kristen Haring*, *Kristen Sparrow*, and *Kristen Lindquist* did not yield any instantcheckmate.com ads (figure 2a, 2c, and 2e), even though the company's database reported having records for all three names and arrest records for *Kristen Sparrow* and *Kristen Lindquist* (figure 2d and 2f).

Searches for *Jill Foley*, *Jill Schneider*, and *Jill James* displayed instantcheckmate.com ads with neutral copy; the word *arrest* did not appear in the ads even though arrest records for all three names appeared in the company's database. Figure 3 shows the ads and criminal reports for these three names appearing on Google.com (figure 1c, 1e) and Reuters.com (figure 1a). Criminal reports came from instantcheckmate.com (figure 1b, 1d, 1f).

FIGURE 2

## Sample Ads and Criminal Reports that Include the First Name,"Kristen"

**A**

Ads by Google

**We Found: Kristen Haring**

1) Contact **Kristen Haring** - Free Info! 2) Current Phone, Address & More.

[www.peoplesmart.com/Kristen](http://www.peoplesmart.com/Kristen)

Search by Phone      Search by Email  
Background Checks      Search by Address  
Public Records      Criminal Records

**Kristen Haring**

Public Records Found For: **Kristen Haring**. Search Now.  
[www.publicrecords.com/](http://www.publicrecords.com/)

**B**

**C**

Ads by Google

**We Found: Kristen Sparrow**

1) Contact **Kristen Sparrow** - Free Info! 2) Current Phone, Address & More.

[www.peoplesmart.com/](http://www.peoplesmart.com/)

Search by Phone      Search by Email  
Background Checks      Search by Address  
Public Records      Criminal Records

**Kristen Sparrow**

Public Records Found For: **Kristen Sparrow**. View Now.  
[www.publicrecords.com/](http://www.publicrecords.com/)

**D**

Name	County and State	Offenses	View Details
1 Kristen Sparrow	CA San Mateo County Superior Court	Criminal Traffic	<a href="#">View Details</a>

**E**

Ads by Google

**Kirsten Lindquist**

Get **Kirsten Lindquist** Find **Kirsten Lindquist**  
[www.ask.com/Kirsten+Lindquist](http://www.ask.com/Kirsten+Lindquist)

**We Found: Kristen Lindquist**

1) Contact **Kristen Lindquist** - Free Info! 2) Current Phone, Address & More.

[www.peoplesmart.com/](http://www.peoplesmart.com/)

Search by Phone      Search by Email  
Background Checks      Search by Address  
Public Records      Criminal Records

**Kristen Lindquist**

Public Records Found For: **Kristen Lindquist**. View Now.  
[www.publicrecords.com/](http://www.publicrecords.com/)

**F**

Name	County and State	Offenses	View Details
1 Kristen Marie Lindquist	Individual NC courts	Criminal Traffic	<a href="#">View Details</a>
2 Kristen Marie Lindquist	NC Admin Office of Courts demographic criminal	Criminal Traffic	<a href="#">View Details</a>
3 Kristen Marie Lindquist	NC Admin Office of Courts demographic criminal	Criminal Traffic	<a href="#">View Details</a>

Finally, we considered a proxy for race associated with these names. Figure 4 shows a racial distinction in the Google images that appear for image searches of *Latanya*, *Latisha*, *Kristen*, and *Jill*, respectively. The faces associated with *Latanya* and *Latisha* tend to be black, while white faces dominate the images of *Kristen* and *Jill*.

Together, these handpicked examples describe the suspected pattern: ads suggesting arrest tend to appear with names associated with blacks, and neutral ads or no ads appear with names associated with whites, regardless of whether the company placing the ad reveals an arrest record associated with the name.



## FIGURE 3

## Sample Ads and Criminal Reports for Names that Include the First Name "Jill"

**A**

Ads by Google

**Located: Jill Foley**  
Information found on **Jill Foley Jill Foley** found in database.  
[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

**Macy's @ Wedding Registry**  
Official Site. Create a Registry or Buy a Wedding Gift at Macy's!  
[www.macys.com/Registry/Wedding](http://www.macys.com/Registry/Wedding)  
macys.com is rated ★★★★★ (82 reviews)

**Dr. Jill Foley**  
View Credentials, Malpractice, Bio, Ratings, Reviews & Background Now!  
[www.lifescript.com/MD](http://www.lifescript.com/MD)

**B**

**JILL FOLEY**  
300 Poppleton Rd  
North Palm Beach, FL, 33408  
DOB: Jul 11, 1941 (84 years old)

**Personal**  
Name, aliases, birthdate, phone numbers, etc.

**Location**  
Current address history and related data, maps, etc.

**Related Persons**  
Family, friends, neighbors, business associates, neighbors, etc.

**Marriage / Divorce**  
Marriage and divorce records on file.

**Criminal History**  
Arrest records, spending issues, judgments, etc.

**Licenses**  
FLA Licenses, USA Licenses, Other Licenses, etc.

**Sex Offenders**  
Sex Offenders living near Jill Foley's primary location.

**Criminal History**  
Rate This Content: ★★★★★  
This section contains possible citations, arrest, and criminal records for the subject of this report. While our database does contain hundreds of millions of arrest records, different counties have different rules regarding what information they will and will not release. We share with you as much information as we possibly can, but a clean state here should not be interpreted as a guarantee that Jill Foley has never been arrested. It simply means that we were not able to locate any matching arrest records in the data that is available to us.

**Possible Matching Arrest Records**

Name	County and State	Offenses	View Details
1 Jill Lee Foley	FL, Pinellas County	Domesticbattery	View Details
2 Jill Lee Foley	FL, Pinellas County	Domesticbattery	View Details

**D**

**JILL SCHNEIDER**  
1507 95th St  
Pawnee, CO, 80419  
DOB: Mar 31, 1989 (35 years old)

**Personal**  
Name, aliases, birthdate, phone numbers, etc.

**Location**  
Current address history and related data, maps, etc.

**Related Persons**  
Family, friends, neighbors, business associates, neighbors, etc.

**Marriage / Divorce**  
Marriage and divorce records on file.

**Criminal History**  
Arrest records, spending issues, judgments, etc.

**Licenses**  
FLA Licenses, USA Licenses, Other Licenses, etc.

**Sex Offenders**  
Sex Offenders living near Jill Schneider's primary location.

**Criminal History**  
Rate This Content: ★★★★★  
This section contains possible citations, arrest, and criminal records for the subject of this report. While our database does contain hundreds of millions of arrest records, different counties have different rules regarding what information they will and will not release. We share with you as much information as we possibly can, but a clean state here should not be interpreted as a guarantee that Jill Schneider has never been arrested. It simply means that we were not able to locate any matching arrest records in the data that is available to us.

**Possible Matching Arrest Records**

Name	County and State	Offenses	View Details
1 Jill E Schneider	WI Admin Office of Courts(CD) Division	Domesticbattery	View Details
2 Jill E Schneider	WI Admin Office of Courts(CD)	Domesticbattery	View Details
3 Jill E Schneider	WI Admin Office of Courts(CD) Division	Domesticbattery	View Details
4 Jill E Schneider	WI Admin Office of Courts(CD)	Domesticbattery	View Details

**E**

Ad related to Jill James

**Located: Jill James**  
[www.instantcheckmate.com/](http://www.instantcheckmate.com/)  
Information found on **Jill James Jill James** found in database.

**F**

**JILL JAMES**  
500 Sycamore Ct  
Cary, NC 27513  
DOB: May 31, 1952 (54 years old)

**Personal**  
Name, aliases, birthdate, phone numbers, etc.

**Location**  
Current address history and related data, maps, etc.

**Related Persons**  
Family, friends, neighbors, business associates, neighbors, etc.

**Marriage / Divorce**  
Marriage and divorce records on file.

**Criminal History**  
Arrest records, spending issues, judgments, etc.

**Licenses**  
FLA Licenses, USA Licenses, Other Licenses, etc.

**Sex Offenders**  
Sex Offenders living near Jill James's primary location.

**Criminal History**  
Rate This Content: ★★★★★  
This section contains possible citations, arrest, and criminal records for the subject of this report. While our database does contain hundreds of millions of arrest records, different counties have different rules regarding what information they will and will not release. We share with you as much information as we possibly can, but a clean state here should not be interpreted as a guarantee that Jill James has never been arrested. It simply means that we were not able to locate any matching arrest records in the data that is available to us.

**Possible Matching Arrest Records**

Name	County and State	Offenses	View Details
1 Jill B James	NC Admin Office of Courts demographic criminal	Domesticbattery	View Details
2 Jill James	NC Admin Office of Courts demographic criminal	Domesticbattery	View Details
3 Jill James	Individual NC courts	Domesticbattery	View Details
4 Jill B James	Individual NC courts	Domesticbattery	View Details
5 Jill Pate James	Individual NC courts	Domesticbattery	View Details
6 Jill Pate James	NC Admin Office of Courts demographic criminal	Domesticbattery	View Details
7 Jill Katy James	NC Admin Office of Courts demographic criminal	Domesticbattery	View Details
8 Jill Katy James	Individual NC courts	Domesticbattery	View Details
9 Jill Rosemond James	NC Admin Office of Courts demographic infraction	Domesticbattery	View Details
10 Jill Rosemond James	NC Admin Office of Courts demographic criminal	Domesticbattery	View Details

## GOOGLE ADSENSE

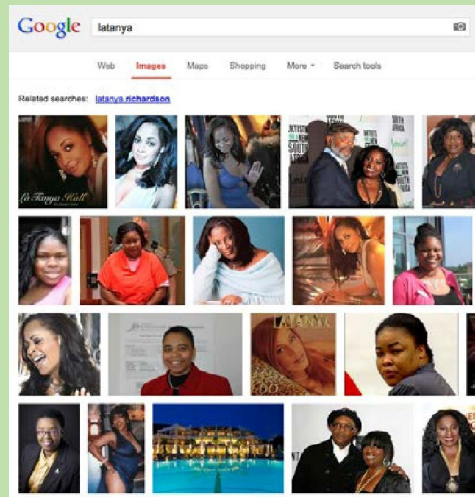
Who generates the ad's text? Who decides when and where an ad will appear? What is the relationship among Google, a news Web site such as Reuters, and Instant Checkmate in the previous examples? An overview of Google AdSense, the program that delivered the ads, explains the links between these companies.

In printed newspapers and magazines, ad space and ad content are fixed. Traditionally, everyone who reads the publication sees the same ad in the same space. Web sites are different. Online ad space, not bound by the same physical limitations, can be dynamic, with ads tailored to the reader's

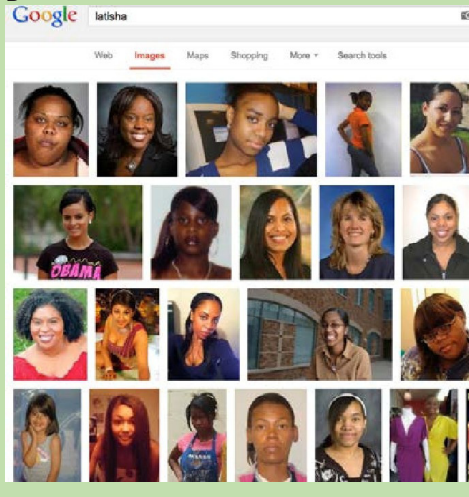
## FIGURE 4

## Sample Face Images on Google.com

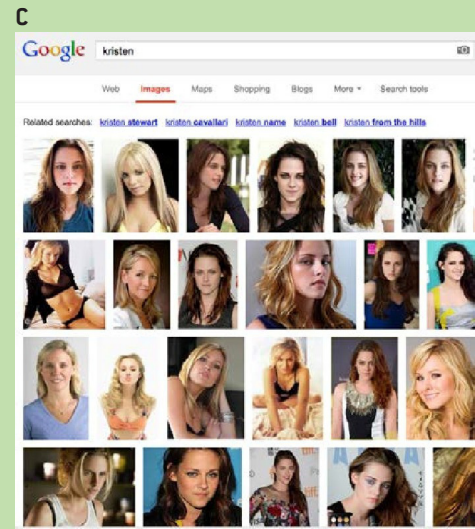
**A**



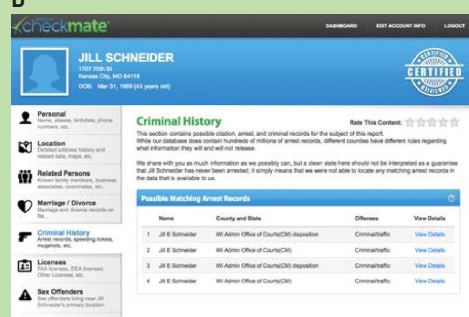
**B**



**C**



**D**



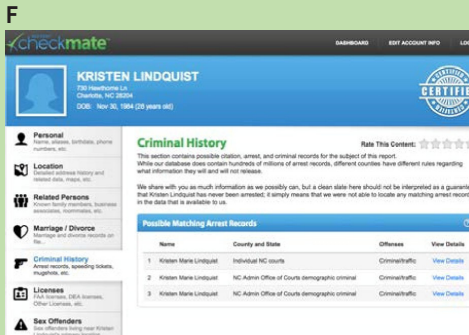
Name	County and State	Offenses	View Details
1. Jill E Schneider	WI Admin Office of Court(23)	Concealment	<a href="#">View Details</a>
2. Jill E Schneider	WI Admin Office of Court(23)	Concealment	<a href="#">View Details</a>
3. Jill E Schneider	WI Admin Office of Court(23)	Concealment	<a href="#">View Details</a>
4. Jill E Schneider	WI Admin Office of Court(23)	Concealment	<a href="#">View Details</a>

**E**

Ad related to Jill James

**Located: Jill James**  
[www.instantcheckmate.com/](http://www.instantcheckmate.com/)  
 Information found on Jill James Jill James found in database.

**F**



Name	County and State	Offenses	View Details
1. Kristen Marie Lindquist	Individual NC courts	Concealment	<a href="#">View Details</a>
2. Kristen Marie Lindquist	NC Admin Office of Courts demographic criminal	Concealment	<a href="#">View Details</a>
3. Kristen Marie Lindquist	NC Admin Office of Courts demographic criminal	Concealment	<a href="#">View Details</a>

search criteria, interests, geographical location, and so on. Any two readers (or even the same reader returning to the same Web site) might view different ads.

Google AdSense is the largest provider of dynamic online advertisements, placing ads for millions of sponsors on millions of Web sites.<sup>9</sup> In the first quarter of 2011, Google earned US\$2.43 billion (\$9.71 billion annualized), or 28 percent of its total revenue, through Google AdSense.<sup>10</sup> Several different advertising arrangements exist, but for simplicity this article describes only those features of Google AdSense specific to the Instant Checkmate ads in question.

When a reader enters search criteria on an enrolled Web site, Google AdSense embeds into

the page of results ads that are believed to be relevant to the search. Figures 1, 2, and 3 show ads delivered by Google AdSense in response to various *firstname lastname* searches.

An advertiser provides Google with search criteria, copies of possible ads to deliver once a match occurs, and a bid of how much the sponsor is willing to pay if a reader clicks the delivered ad. (This article conflates two interacting Google programs: Google AdWords allows advertisers to specify search criteria, ad text, and bids; and Google AdSense delivers the ads to host sites.) Google operates a realtime auction across bids for the same search criteria, computing an overall “quality score” to use as the basis for the auction. The quality score includes many factors such as the past performance of the ad and characteristics of the company’s Web site.<sup>10</sup> The ad with the highest quality score appears first, the second-highest second, and so on, and Google may elect not to show any ad if it considers the bid too low or if showing the ad exceeds a threshold (e.g., a maximum account total for the sponsor). The Instant Checkmate ads in figures 1, 2, and 3 often appeared first among ads, implying Instant Checkmate had the highest quality score.

A Web-site owner that wants to “host” online ads enrolls in AdSense and modifies the Web site to include special software that sends information about the current reader (e.g., search criteria) to Google; in exchange, the Web site receives corresponding ads from Google. The displayed ads have the banner “Ads by Google” when they appear on sites other than Google.com. For example, Reuters.com is an AdSense host, and entering *Latanya Sweeney* in the search bar generated a new Web page with ads delivered by Google, bearing the banner “Ads by Google” (figure 1c).

There is no cost associated with displaying an ad, but if the user actually clicks the ad, the sponsor pays the bid price. This may be as little as a few pennies, and the amount is split between Google and the host. Clicking the *Latanya Sweeney* ad on Reuters.com (figure 1c) would cause Instant Checkmate to pay its bid to Google, and Google would split the payment with Reuters.

#### SEARCH CRITERIA

What search criteria did Instant Checkmate specify? Are ads randomly delivered? Do ads rely only on the first name? Will ads be delivered for made-up names? Google AdSense provides answers to these questions. Ads displayed on Google.com allow users to learn why a specific ad appeared. Clicking the circled “i” in the ad banner (e.g., figure 1c) leads to a Web page explaining the ads. Doing so for ads in figures 1 and 3 reveals that the ads appeared because the search criteria associated with the bid matched the exact first- and last-name combination searched. Because a company presumably bids on records it sells, the names would likely be the first and last names of real people.

This means that the search criteria associated with these ads have to consist of both first and last names, and the names should belong to real people.

The next steps describe the systematic construction of a list of racially associated first and last names for real people to use as search criteria. Instant Checkmate is not presumed to have used such a list in placing bids, nor Google in delivering ads. Rather, the list provides a qualified sample of racially associated names to use in testing ad-delivery systems.

#### BLACK- AND WHITE-IDENTIFYING NAMES

Black-identifying and white-identifying first names occur with sufficiently higher frequency in one race than the other.

In 2003 Marianne Bertrand and Sendhil Mullainathan of the NBER (National Bureau of Economic



Research) did a field experiment in which they provided resumes to job ads that were virtually identical, except that some of the resumes had black-identifying names and others had white-identifying names.<sup>2</sup> Their job discrimination study showed significant discrimination against black names: white names received 50 percent more callbacks for interviews, even though the resumes otherwise had identical qualifications.

The study used a correlation of names given to black and white babies in Massachusetts between 1974 and 1979, defining black-identifying and white-identifying names as those that have the highest ratio of frequency in one racial group to frequency in the other racial group.

In the popular book *Freakonomics* (William Morrow, 2006), Steven Levitt and Stephen Dubner report the top 20 whitest- and blackest-identifying girls' and boys' names. The list comes from earlier work by Levitt and Roland Fryer, which shows a pattern change in the way blacks named their children starting in the 1970s, which they correlate with the Black Power Movement.<sup>7</sup> They postulate that the movement influenced how blacks perceived their identities, and they give as evidence that before the movement, names given to black and white children were not distinctly different, but after the movement distinctly black names emerged.

Similar to the job discrimination study, the list used by Fryer and Levitt was compiled from names given to black and white children recorded in California birth records from 1961-2000 (more than 16 million births).

To test methods of ad delivery, we combined the lists from these prior studies and added two black female names, *Latanya* and *Latisha*. Table 1 lists the names used here, consisting of eight for each of the categories: white female, black female, white male, and black male from the Bertrand and Mullainathan job discrimination study (first row in table 1); and the first eight names for each category from the Fryer and Levitt work (second row in table 1). Emily, a white female name, Ebony, a black female name, and Darnell, a black male name, appear in both rows. The third row includes the observation shown in figure 4. Removing duplicates leaves a total of 63 distinct first names.

TABLE 1. Black-identifying and white-identifying first names

White Female	Black Female	White Male	Black Male
Allison	Aisha	Brad	Darnell
Anne	Ebony	Brendan	Hakim
Carrie	Keisha	Geoffrey	Jermaine
Emily	Kenya	Greg	Kareem
Jill	Latonya	Brett	Jamal
Laurie	Lakisha	Jay	Leroy
Kristen	Latoya	Matthew	Rasheed
Meredith	Tamika	Neil	Tremayne
Molly	Imani	Jake	DeShawn
Amy	Ebony*	Connor	DeAndre
Claire	Shanice	Tanner	Marquis
Emily*	Aaliyah	Wyatt	Darnell*
Katie	Precious	Cody	Terrell
Madeline	Nia	Dustin	Malik
Katelyn	Deja	Luke	Trevon
Emma	Diamond	Jack	Tyrone
	Latanya		
	Latisha		

## FULL NAMES OF REAL PEOPLE

Having a list of racially associated first names is a start, but testing ad delivery requires a real person's first and last name (full name). Web searches provide a means of locating and harvesting full names by: (1) sampling names of professionals appearing on the Web; and (2) sampling names of people active on social media sites and blogs (netizens).

Professionals often have their own Web sites or have biographical information appearing on institutional Web sites, listing titles and positions and describing prior accomplishments and current activities. Several professions, such as research, medicine, law, and business, often have degree designations (e.g., PhD, MD, JD, or MBA) associated with people in that profession. A Google search for a first name and a degree designation can yield lists of people having that first name and degree. These kinds of searches can harvest a sample of full names of professionals with racially associated first names.

The next step is to visit the Web page associated with each full name, and if an image is discernible, record whether the person appears black, white, or other. Each Web page visited should be archived to preserve images and content.

Here are two examples from my ad-delivery test. A Google search for *Ebony PhD* revealed links for real people having *Ebony* as a first name—specifically, *Ebony Bookman*, *Ebony Glover*, *Ebony Baylor*, and *Ebony Utley*. I harvested the full names appearing on the first three pages of search results, using searches with other professional endings such as *JD*, *MD*, or *MBA* as needed to find at least 10 full names for *Ebony*. Clicking on the link associated with *Ebony Glover* provided more information about her, including an image.<sup>8</sup> The *Ebony Glover* in this study appeared black.

Similarly, search results for *Jill PhD* listed professionals whose first name is *Jill*. Visiting links yielded Web pages with more information about each person. For example, *Jill Schneider's* Web page had an image showing that she is white.<sup>14</sup>

Harvesting names of netizens is similar but simpler than harvesting names of professionals. PeekYou searches were used to harvest a sample of full names of netizens who have racially associated first names. The Web site peekyou.com compiles online and offline information on individuals—thereby connecting residential information with Facebook and Twitter users, bloggers, and others—and assigns its own rating for the size of each person's online footprint. Search results from peekyou.com list people with the highest score first, and include an image of the person. Celebrities and public figures tend to be listed first, having the highest PeekYou scores, followed by bloggers, tweeters, and the rest.

A PeekYou search for *Ebony* found *Ebony Small*, *Ebony Cams*, *Ebony King*, *Ebony Springer*, and *Ebony Tan*. A PeekYou search for *Jill* found *Jill Christopher*, *Jill Spivack*, *Jill English*, *Jill Pantozzi*, and *Jill Dobson*. After harvesting these and other full names, I reported the race of the person if discernible.

Using the approach just described, I harvested 2,184 racially associated full names of people with an online presence from September 24 through October 22, 2012. Using the images associated with those names, I was able to confirm that the racially associated first names were predictive of race.<sup>15</sup> Most images associated with black-identifying names were of black people (88 percent), and an even greater percentage of images associated with white-identifying names were of white people (96 percent).

Black names and white names were examined separately as predictors of race. The results showed that 490 images of blacks had black-associated first names, and 68 did not; 18 images of blacks had

white first names; 852 had neither black first names nor images of blacks. Similarly, 831 images of whites had white first names, 50 images of whites did not have white first names; 39 had white first names but nonwhite images, and 508 had neither white first names nor images of whites.

Google searches of first names and degree designations were not as productive as first name lookups on PeekYou. On Google, the white male names *Cody*, *Connor*, *Tanner*, and *Wyatt* retrieved results with those as last names rather than first names; the black male name *Kenya* was confused with the country; and the black names *Aaliyah*, *Deja*, *Diamond*, *Hakim*, *Malik*, *Marquis*, *Nia*, *Precious*, and *Rasheed* retrieved fewer than 10 full names. Only *Diamond* posed a problem with PeekYou searches, seemingly confused with other online entities. *Diamond* was therefore excluded from further consideration.

Some black first names had perfect predictions (100 percent): *Aaliyah*, *DeAndre*, *Imani*, *Jermaine*, *Lakisha*, *Latoya*, *Malik*, *Tamika*, and *Trevon*. The worst predictors of blacks were *Jamal* (48 percent) and *Leroy* (50 percent). Among white first names, 12 of 31 names made perfect predictions: *Brad*, *Brett*, *Cody*, *Dustin*, *Greg*, *Jill*, *Katelyn*, *Katie*, *Kristen*, *Matthew*, *Tanner*, and *Wyatt*; the worst predictors of whites were *Jay* (78 percent) and *Brendan* (83 percent). These findings strongly support the use of these names as racial indicators in this study.

Sixty-two full names appeared in the list twice even though the people were not necessarily the same. No name appeared more than twice. Overall, Google and PeekYou searches tended to yield different names.

#### AD DELIVERY

With this list of names suggestive of race, I was ready to test which ads appear when these names are searched. To do this, I examined ads delivered on two sites, Google.com and Reuters.com, in response to searches of each full name, once at each site. The browser's cache and cookies were cleared before each search, and copies of Web pages received were preserved. Figures 1, 2, 3, 6, and 7 provide examples.

From September 24 through October 23, 2012, I searched 2,184 full names on Google.com and Reuters.com. The searches took place at different times of day, on different days of the week, with different IP and machine addresses operating in different parts of the United States using different browsers. I manually searched 1,373 of the names and used automated means<sup>17</sup> for the remaining 812 names. Here are 10 observations.

**1. The ads were respectfully displayed, without clutter.** We have all seen Web pages where ads get in the way, dominating the page or being so closely woven into the page that you cannot distinguish the ads from the content. That's not the case here. No more than three ads ever appeared for a search on either Google.com or Reuters.com. No company's ad was listed more than once on a page, and the ads appeared in a single set within a rectangular area in the margins. Google and Reuters are respected sources of information, and displayed in this manner, the ads did nothing to take away from the Web sites; conversely, the sites and respectful placement of ads may even exalt the ads.

**2. Far fewer ads appeared on Google.com than Reuters.com**—about five times fewer, even when examining up to three pages of search results on Google.com. When ads did appear on Google.com, typically only one ad showed, compared with three ads routinely appearing on Reuters.com. This suggests Google may be sensitive to the number of ads appearing on Google.com.

**3. Of 5,337 ads captured, 78 percent were for government-collected information (public records)**

**about the person whose name was searched.** Public records in the United States often include a person's address, phone number, criminal history, and professional and business licenses, though specifics vary among states. Of the more than 2,000 names searched, 78 percent had at least one ad for public records about the person being searched. Ads to buy a person's public record appeared for almost any name searched, but they came up on Reuters.com much more often than on Google.com.

**4. Four companies had more than half of all the ads captured.** These companies were Instant Checkmate, PublicRecords (which is owned by Intelius), PeopleSmart, and PeopleFinders, and all their ads were selling public records. Instant Checkmate ads appeared more than any other: 29 percent of all ads. Ad distribution was different on Google's site; Instant Checkmate still had the most ads (50 percent), but Intelius.com, while not in the top four overall, had the second most ads on Google.com. These companies dominate the advertising space for online ads selling public records.

**5. Instant Checkmate ads dominated the topmost ad position.** They occupied that spot in almost half of all searches on Reuters.com. The next closest, PublicRecords.com, rarely had the topmost spot, but most frequently appeared in the second and third positions. Appearing as the first ad so often suggests that, in general, Instant Checkmate offers Google more money or has higher quality scores than do its competitors.

**6. Ads for public records on a person appeared more often for those with black-associated names than white-associated names, regardless of company.** PeopleSmart ads appeared disproportionately higher for black-identifying names—41 percent as opposed to 29 percent for white names. PublicRecords ads appeared 10 percent more often for black first names than for white. Instant Checkmate ads displayed only slightly more often for black-associated names (2 percent difference). This is one of those interesting findings that spawn the question: Public records contain information on everyone, so why more ads for black-associated names?

**7. Instant Checkmate had the largest percentage of ads in virtually every first-name category, except for Kristen, Connor, and Tremayne.** For those names, Instant Checkmate had uncharacteristically fewer ads (less than 25 percent). PublicRecords had ads for 80 percent of names beginning with *Tremayne*, compared with only 23 percent for Instant Checkmate. Similarly, for *Connor*, PublicRecords had 80 percent compared with 20 percent for Instant Checkmate, and for *Kristen* it was 58 percent PublicRecords versus 16 percent Instant Checkmate. Why the underrepresentation in these first names? Did Instant Checkmate avoid these names for some reason? Do these undercounts show a glitch? During a conference call with company's representatives, they asserted that Instant Checkmate gave the same ad text to Google for groups of last names (not first names).

**8. Almost all ads for public records included the name of the person, making each ad virtually unique, but beyond personalization, the ad templates showed little variability.** The only exception was Instant Checkmate. For example, almost all PeopleFinder ads appearing on Reuters.com used the same personalized template ("We found *fullname*. Current Address, Phone and Age. Find *fullname*, Anywhere," where the person's first and last name replaces *fullname*). PublicRecords used five templates and PeopleSmart seven, but Instant Checkmate used 18 different ad templates on Reuters.com. Figure 5 enumerates ad texts and frequencies for all four companies (replace *fullname* with the person's first and last name).

Only Instant Checkmate ads used the word *arrest*, which appeared in eight of its 18 ad templates found on Reuters.com. While Instant Checkmate's competitors—PeopleSmart, PublicRecords, and PeopleFinders—also sell criminal history information, none of their ads included the word *arrest* or *arrested*.



## FIGURE 5

## Templates for Ads for Public Records on Reuters

<b>instantcheckmate</b>	<b>Peoplesmart</b>
382 <b>Located: <u>fullname</u></b> Information found on <u>fullname</u> <u>fullname</u> found in database.	7 <b>We found: <u>fullname</u></b> 1) Get <u>firstname</u> 's Background Report 2) Contact info & More -try Free!
2 <b>Located: The Person</b> Information found on them Person found in database.	87 <b>We found: <u>fullname</u></b> 1) Get Aisha's Background Report 2) Current Contact Info - Try Free!
96 <b>We found <u>fullname</u></b> Search Arrests, Address, Phone, etc. Search records for <u>fullname</u> .	105 <b>We found: <u>fullname</u></b> 1) Contact <u>fullname</u> -Free Info! 2) Current Address, Phone & More.
4 <b>We found Them</b> Search Arrests, Address, Phone, etc. Search records for <u>fullname</u> .	348 <b>We found: <u>fullname</u></b> 1) Contact <u>fullname</u> -Free Info! 2) Current Phone, Address & More.
40 <b>Background of <u>fullname</u></b> Search Instant Checkmate for the Records of <u>fullname</u>	1 <b>We found <u>firstname</u></b> Get <u>firstname</u> in CA's Email, Address, Phone, Public Records & More Easy!
9 <b>Background of Anyone</b> Search Instant Checkmate for the Records of <u>fullname</u>	1 <b>We found <u>firstname</u> In <u>lastname</u></b> 1)Get <u>firstname</u> 's Info – Try Now! 2)Current Phone, Address & More.
17 <b><u>fullname</u>'s Records</b> 1) Enter Name and State. 2) Access Full Background Checks Instantly.	1 <b>Looking For <u>fullname</u>?</b> Get <u>fullname</u> 's Phone, Email Address, Public Records & More Now!
3 <b>Anyone's Records</b> 1) Enter Name and State. 2) Access Full Background Checks Instantly.	<b>Publicrecords</b>
195 <b><u>fullname</u>: Truth</b> Arrests and Much More. Everything About <u>fullname</u>	570 <b><u>fullname</u></b> Public Records Found For: <u>fullname</u> . View now.
67 <b><u>fullname</u> Truth</b> Looking for <u>fullname</u> ? Check <u>fullname</u> 's Arrests	128 <b><u>fullname</u></b> Public Records Found For: <u>fullname</u> . Search now.
176 <b><u>fullname</u>, Arrested?</b> 1) Enter Name and State. 2) Access Full Background Checks Instantly.	13 <b>Records: <u>fullname</u></b> Database of all <u>lastname</u> 's in the Country. Search now.
2 <b>Uh Oh, Arrested?</b> 1) Enter Name and State. 2) Access Full Background Checks Instantly.	2 <b>Fullname Info</b> View Contact Information For Free Quick & Easy Search Results!
1 <b>Found: <u>fullname</u></b> We have the story on <u>fullname</u> <u>fullname</u> 's arrests, relatives, etc.	56 <b><u>fullname</u></b> We have Public Records For: <u>fullname</u> . Search Now.
3 <b>Fullname - Found</b> Learn the truth about <u>fullname</u> Check <u>fullname</u> 's arrests & more.	<b>Peoplefinders</b>
4 <b>Research <u>fullname</u></b> We have details on <u>fullname</u> . <u>fullname</u> 's full background & info.	523 <b>We found <u>fullname</u></b> Current Address, Phone and Age. Find <u>fullname</u> , Anywhere.
55 <b><u>fullname</u> Located</b> Background Check, Arrest Records, Phone, & Address. Instant, Accurate	8 <b>We found <u>fullname</u></b> 1)Get Phone/ Address/ Age Instantly! 2) Find Anyone, Anywhere for Free.
62 <b>Looking for <u>fullname</u>?</b> Comprehensive Background Report and More on <u>fullname</u>	2 <b>Find <u>fullname</u></b> Get current and past addresses and phone numbers. Instant results!
8 <b>Looking for People in the US?</b> Comprehensive Background Report and More on <u>fullname</u>	1 <b>We Found Them for Free</b> Current Address, Phone and Age. Find <u>fullname</u> Anywhere.

9. A greater percentage of Instant Checkmate ads using the word *arrest* appeared for black-identifying first names than for white first names. More than 1,100 Instant Checkmate ads appeared on Reuters.com, with 488 having black-identifying first names; of these, 60 percent used *arrest* in the ad text. Of the 638 ads displayed with white-identifying names, 48 percent used *arrest*. This difference is statistically significant, with less than a 0.1 percent probability that the data can be explained by chance (chi-square test:  $\chi^2(1)=14.32$ ,  $p < 0.001$ ). The EEOC's and U.S. Department of Labor's adverse impact test for measuring discrimination is 77 in this case, so if this were an employment situation, a charge of discrimination might result. (The adverse impact test uses the ratio of neutral ads, or 100 minus the percentages given, to compute disparity:  $100-60=40$  and  $100-48=52$ ; dividing 40 by 52 equals 77.)

The highest percentage of neutral ads (where the word *arrest* does not appear in the ad text) on Reuters.com were those for *Jill* (77 percent) and *Emma* (75 percent), both white-identifying names. Names receiving the highest percentage of ads with *arrest* in the text were *Darnell* (84 percent), *Jermaine* (81 percent), and *DeShawn* (86 percent), all black-identifying first names. Some names

appeared counter to this pattern: *Dustin*, a white-identifying name, generated *arrest* ads in 81 percent of searches; and *Imani*, a black-identifying name, resulted in neutral ads in 75 percent of searches.

**10. Discrimination results on Google's site were similar, but, interestingly, ad text and distributions were different.** Instant Checkmate ads appearing on Google.com often used different ad text than those on Reuters.com. While the same neutral and *arrest* ads that were dominant on Reuters.com also appeared frequently on Google.com, Instant Checkmate ads on Google included an additional 10 templates, all using the word *criminal* or *arrest*. These new templates appeared in about 20 percent of the Instant Checkmate ads on Google.

More than 400 Instant Checkmate ads appeared on Google, and 90 percent of these were suggestive of *arrest*, regardless of race. Together, these last two findings underscore other differences between ads appearing on Google's own site and those delivered by Google AdSense to Reuters. Ad text was different. Ads with the word *criminal* and not *arrest* appeared only on Google's site, and ads using either *arrest* or *criminal* appeared much more often for both races on Google.com.

Still, on Google's own site, a greater percentage of Instant Checkmate ads suggestive of arrest displayed for black-associated first names than for white-associated names. Of the 366 ads that appeared for black-identifying names, 92 percent were suggestive of arrest. Far fewer ads displayed for white-identifying names (66 total), but 80 percent were suggestive of arrest. This difference in the ratios 92 and 80 is statistically significant, with less than a 1 percent probability that the data can be explained by chance (chi-square test:  $\chi^2(1)=7.71$ ,  $p < 0.01$ ). The EEOC's adverse impact test for measuring discrimination is 40 percent, so in an employment situation, a charge of discrimination might result. (The adverse impact test gives  $100-92=8$  and  $100-80=20$ ; dividing 8 by 20 gives 40 percent.)

A greater percentage of Instant Checkmate ads with the word *arrest* in ad text appeared for black-identifying first names than for white-identifying first names within professional and netizen subsets, too.

This study started with the hypothesis that no difference exists in the delivery of ads suggestive of an arrest record based on searches of racially associated names. The findings reject this. A greater percentage of ads using *arrest* in their text appeared for black-identifying first names than for white-identifying first names in searches on Reuters.com, Google.com, and in subsets of the sample. On Reuters.com, which hosts Google AdSense ads, a black-identifying name was 25 percent more likely to generate an ad suggestive of an arrest record.

### THREE ADDITIONAL OBSERVATIONS

The people behind the names used in this study are diverse. Political figures included State Representatives Aisha Braveboy (*arrest* ad) and Jay Jacobs (neutral ad) of Maryland; Jill Biden (neutral ad), wife of U.S. Vice President Joe Biden; and Claire McCaskill, whose campaign ad for the U.S. Senate in Missouri appeared alongside an Instant Checkmate ad using the word *arrest* (figure 6). Names mined from academic Web sites included graduate students, researchers, administrators, staff, and accomplished academics, such as Amy Gutmann, president of the University of Pennsylvania and chair of the U.S. Presidential Commission for the Study of Bioethical Issues. Dustin Hoffman (*arrest* ad) was among the celebrity names used. A smorgasbord of athletes appeared, from local to national fame (assorted neutral and *arrest* ads). The youngest person whose name was used in the study was a missing 11-year-old black girl.

## FIGURE 6

## Example Ads Displayed in Response to Search of “Claire McCaskill”



More than 1,100 of the names harvested for this study were from PeekYou, with scores estimating the name’s overall presence on the Web. As expected, celebrities get the highest scores of 10s and 9s. Only four names used here had a PeekYou score of 10, and 12 had a score of 9, including Dustin Hoffman. Only two ads appeared for these high-scoring names; an abundance of ads appeared across the remaining spectrum of PeekYou scores. It seems likely that the bid price needed to get an ad placed first is greater for more well-known and popular names with higher PeekYou scores. Knowing that very few high-scoring people were in the study and that ads appeared across the full spectrum of PeekYou scores reduces concern about variations in bid prices.

Different Instant Checkmate ads sometimes appeared for the same person. About 200 names had Instant Checkmate ads on both Reuters.com and Google.com, but only 42 of these names received the same ad. The other 82 percent of names received different ads across the two sites. Search results on Reuters.com for the 62 duplicate names that appeared in the study showed different ads for 37 of them, the same ad for seven, and no ad for 18. At most, three distinct ads appeared across Reuters.com and Google.com for the same name. Figure 7 shows the assortment of ads appearing for *Latonya Evans* and *Latisha Smith*. Having different possible ad texts for a name reminds us that while Instant Checkmate provided the ad texts, Google’s technology selected among the possible texts in deciding which to display. In Figure 7, each name had ads both suggestive of arrest and not, though they both had more ads suggestive of arrest than not.

## MORE ABOUT THE PROBLEM

Why is this discrimination occurring? Is Instant Checkmate, Google, or society to blame? We don’t yet know, but navigating the terrain requires further information about the inner workings of Google AdSense. Google understands that an advertiser may not know which ad copy will work best, so the advertiser may provide multiple templates for the same search string, and the “Google algorithm” learns over time which ad text gets the most clicks from viewers. It does this by assigning weights (or probabilities) based on the click history of each ad. At first, all possible ad texts are weighted the same and are presumed equally likely to produce a click. Over time, as people click one version of an ad more often than others, the weights change, so the ad text getting the most clicks

## FIGURE 7

## Examples of Different Ad Copy Appearing for Different Names

**A**

Ads by Google

[Latonya Evans, Arrested?](#)

1) Enter Name and State. 2) Access Full Background Checks Instantly.

[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

**A**

Ads by Google

[Latonya Evans's Records](#)

1) Enter Name and State. 2) Access Full Background Checks Instantly.

[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

**A**

Ad related to Latonya Evans ⓘ

[Latonya Evans's Records](#)

[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

Did you know **Latonya Evans's** criminal history is searchable?

**B**

Ads by Google

[Latisha Smith Located](#)

Background Check, Arrest Records, Phone, & Address. Instant, Accurate

[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

**B**

Ads related to Latisha Smith ⓘ

[Latisha Smith, Arrested?](#)

[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

1) Enter Name and State. 2) Access Full Background Checks Instantly.

**B**

Ads by Google

[Latisha smith: Truth](#)

Arrests and Much More. Everything About Latisha smith

[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

**B**

Ad related to Latisha Smith ⓘ

[Latisha Smith, Arrested? - 1\) Enter Name and State.](#)

[www.instantcheckmate.com/](http://www.instantcheckmate.com/)

2) Access Full Background Checks Instantly.

eventually displays more frequently. This approach aligns the financial interests of Google, as the ad deliverer, with the advertiser.

Did Instant Checkmate provide ad templates suggestive of arrest disproportionately to black-identifying names? Or did Instant Checkmate provide roughly the same templates evenly across racially associated names but users clicked ads suggestive of arrest more often for black-identifying names? As mentioned earlier, during a conference call with the founders of Instant Checkmate and their lawyer, the company's representatives asserted that Instant Checkmate gave the same ad text to Google for groups of last names (not first names) in its database; they expressed no other criteria for name and ad selection.

Google uses cloud-caching strategies to deliver ads quickly. Might these strategies create a bias toward templates previously loaded in the cloud cache? Is there a combination effect?

This study is a start, but more research is needed. To preserve research opportunities, I captured additional results for 50 hits on 2,184 names across 30 Web sites serving Google Ads to learn the underlying distributions of ad occurrences per name. While analyzing the data may prove illuminating, in the end the basic message presented in this study does not change: there is discrimination in delivery of these ads.

## TECHNICAL SOLUTIONS

How can technology solve this problem? One answer is to change the quality scores of ads to



discount for unwanted bias. The idea is to measure realtime bias in an ad's delivery and then adjust the weight of the ad accordingly at auction. The general term for Google's technology is *ad exchange*. This approach integrates seamlessly into the way ad exchanges operate, allowing minimal modifications to harmonize ad deliveries with societal norms; it generalizes to other ad exchanges (not just Google's); and, finally, it works regardless of the cause of the discrimination—advertiser bias in placing ads or societal bias in selecting ads.

Discrimination, however, is at the heart of online advertising. Differential delivery is the very idea behind it. For example, if young women with children tend to purchase baby products and retired men with bass boats tend to purchase fishing supplies, and you know the viewer is one of these two types, then it is more efficient to offer ads for baby products to the young mother and fishing rods to the fisherman, not the other way around.

On the other hand, not all discrimination is desirable. Societies have identified groups of people to protect from specific forms of discrimination. Delivering ads suggestive of arrest much more often for searches of black-identifying names than for white-identifying names is an example of unwanted discrimination, according to American social and legal norms. This is especially true because the ads appear regardless of whether actual arrest records exist for the names in the company's database.

The good news is that we can use the mechanics and legal criteria described earlier to build technology that distinguishes between desirable and undesirable discrimination in ad delivery. Key components are: (1) identifying affected groups; (2) specifying the scope of ads to assess; (3) determining ad sentiment; and (4) testing for adverse impact.

**1. Identifying affected groups.** A set of predicates can be defined to identify members of protected and comparison groups. Given an ad's search string and text, a predicate returns *true* if the ad can impact the group that is the subject of the predicate and returns *false* otherwise. Statistics of baby names can identify first names for constructing race and gender groups and last names for grouping some ethnicities. Special word lists or functions that report degree of membership may be helpful for other comparisons.

In this study, ads appeared on searches of full names for real people, and first names assigned to more black or white babies formed groups for testing. These *black* and *white* predicates evaluate to *true* or *false* based on the first name of the search string.

**2. Specifying the scope of ads to assess.** The focus should be on those ads capable of impacting a protected group in a form of discrimination prohibited by law or social norm. Protection typically concerns the ability to give or withhold benefits, facilities, services, employment, or opportunities. Instead of lumping all ads together, it is better to use search strings, ad texts or products, or URLs that display with ads to decide which ads to assess.

This study assessed search strings of first and last names of real people, ads for public records, and ads having a specific display URL ([instantcheckmate.com](http://instantcheckmate.com)), the latter being the most informative because the adverse ads all had the same display URL.

Of course, the audience for the ads is not necessarily the people who are the subjects of the ads. In this study, the audience is a person inquiring about the person whose name is the subject of the ad. This distinction is important when thinking about the identity of groups that might be impacted by an ad. Group membership is based on the ad's search string and text. The audience may resonate more with a distinctly positive or negative characterization of the group.

**3. Determining ad sentiment.** Originally associated with summarizing product and movie reviews, sentiment analysis is an area of computer science that uses natural-language processing

and text analytics to determine the overall attitude of a text.<sup>13</sup> Sentiment analysis can measure whether an ad's search string and accompanying text have positive, negative, or neutral sentiment. A literature search does not find any prior application to online ads, but a lot of research has been done assessing sentiment in social media (sentiment140.com, for example, reports the sentiment of tweets, which like advertisements have limited words).

In this study ads containing the word *arrest* or *criminal* were classified as having negative sentiment, and ads without those words were classified as neutral.

**4. Testing for adverse impact.** Consider a table where columns are comparative groups, rows are sentiment, and values are the number of ad impressions (the number of times an ad appears, whether or not it is clicked). Ignore neutral ads. Comparing the percentage of ads having the same positive or negative sentiment across groups reveals the degree to which one group may be impacted more or less by the ad's sentiment. A chi-square test can determine statistical significance, and the adverse impact test used by the EEOC and the U.S. Department of Labor can indicate whether in some circumstances the impact may lead to legal risks.

In this study the groups are black and white, and the sentiments are negative and neutral. Table 2 shows a summary chart. Of the 488 ads that appeared for the black group, 291 (or 60 percent) had negative sentiment. Of the 638 ads displayed for the white group, 308 (or 48 percent) had negative sentiment. The difference is statistically significant ( $\chi^2(1)=14.32$ ,  $p < 0.001$ ) and has an adverse impact measure of 40/52, or 77 percent.

An easy way of incorporating this analysis into an ad exchange is to decide which bias test is critical (e.g., statistical significance or the adverse impact test) and then factor the test result into the quality score for the ad at auction. For example, if we were to modify the ad exchange not to display any ad with an adverse impact score of less than 80, which is the EEOC standard, then arrest ads for blacks would sometimes appear, but would not be overly disproportionate to such ads for whites, regardless of advertiser or click bias.

Though this study served as an example throughout, the approach generalizes to many other forms of discrimination and combats other ways of fostering discrimination.

Suppose female names tend to get neutral ads such as "Buy now," while male names tend to get positive ads such as "Buy now. 50% off!" Or suppose black names tend to get neutral ads such as "Looking for Ebony Jones," while white names tend to get positive ads such as "Meredith Jones. Fantastic!" Then the same analysis would suppress some occurrences of the positive ads so as not to foster a discriminatory effect.

This approach does not stop the appearance of negative ads for a store placed by a disgruntled customer or ads placed by competitors on brand names of the competition, unless these are deemed to be protected groups.

TABLE 2. Assessment of negative and neutral ads for black and white groups

	Black		White	
Negative	291	60%	308	48%
Neutral	197	40%	330	52%
Positive	--		--	
Total	488		638	

Nonprotected marketing discrimination can continue even to protected groups. For example, suppose search terms associated with blacks tend to get neutral ads for some music artists, while those associated with whites tend to get neutral ads for other music artists. All ads would appear regardless of the disproportionate distribution because the ads would not be subject to suppression.

As a final example, this approach allows everyone to be negatively impacted as long as the impact is roughly the same. Suppose all ads for public records on all names, regardless of race, were equally suggestive of arrest and had almost the same number of impressions; then no ads suggestive of arrest would be suppressed.

Computer scientist Cynthia Dwork and her colleagues have already been working on algorithms that assure racial fairness.<sup>4</sup> Their general notion is to make sure similar groups receive similar ads in proportions consistent with the population. Utility is the critical concern with this direction because not all forms of discrimination are bad, and unusual and outlier ads could be unnecessarily suppressed. Still, their research direction looks promising.

In conclusion, this study demonstrates that technology can foster discriminatory outcomes, but it also shows that technology can thwart unwanted discrimination.

#### ACKNOWLEDGMENTS

The author thanks Ben Edelman, Claudine Gay, Gary King, Annie Lewis, and weekly Topics in Privacy participants (David Abrams, Micah Altman, Merce Crosas, Bob Gelman, Harry Lewis, Joe Pato, and Salil Vadhan) for discussions; Adam Tanner for first suspecting a pattern; Diane Lopez and Matthew Fox in Harvard's Office of the General Counsel for making publication possible in the face of legal threats; and Sean Hooley for editorial suggestions. Data from this study is available at [foreverdata.org](http://foreverdata.org) and the IQSS Dataverse Network. Supported in part by NSF grant CNS-1237235 and a gift from Google, Inc.

#### REFERENCES

1. Barker R. 2003. *The Social Work Dictionary* (5th ed.). Washington, DC: NASW Press.
2. Bertrand, M., Mullainathan, S. 2003. Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination. NBER Working Paper No. 9873; .
3. *Central Hudson Gas & Electric Corp. v. Public Service Commission of New York*. 1980. Supreme Court of the United States, 447 U.S. 557.
4. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R. 2011. Fairness through awareness. arXiv:1104.3913 [cs.CC]; <http://arxiv.org/abs/1104.3913>.
5. Equal Employment Opportunity Commission. 2012. Consideration of arrest and conviction records in employment decisions under Title VII of the Civil Rights Act of 1964. Washington, DC. 915.002; [http://www.eeoc.gov/laws/guidance/arrest\\_conviction.cfm](http://www.eeoc.gov/laws/guidance/arrest_conviction.cfm).
6. Equal Employment Opportunity Commission. 1978. Uniform guidelines on employee selection procedures. Washington, DC.
7. Fryer, R., Levitt, S. 2004. The causes and consequences of distinctively black names. *The Quarterly Journal of Economics* 59(3); <http://pricetheory.uchicago.edu/levitt/Papers/FryerLevitt2004.pdf>.
8. Glover, E; <http://www.physiology.emory.edu/FIRST/ebony2.htm> (archived at <http://foreverdata.org/onlineads>).
9. Google AdSense; <http://google.com/adsense>.

10. Google. Google announces first quarter 2011 financial results; [http://investor.google.com/earnings/2011/Q1\\_google\\_earnings.html](http://investor.google.com/earnings/2011/Q1_google_earnings.html).
11. Harris, P., Keller, K. 2005. Ex-offenders need not apply: the criminal background check in hiring decisions. *Journal of Contemporary Criminal Justice* 21(1): 6-30.
12. Panel on Methods for Assessing Discrimination, National Research Council. 2004. Measuring racial discrimination. Washington, DC: National Academy Press.
13. Pang, B., Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.
14. Schneider, J. <http://www.lehigh.edu/bio/jill.html> (Archived at <http://foreverdata.org/onlineads>).
15. Sweeney, L. 2013. Discrimination in online ad delivery. (For detailed results and analysis, see full technical report archived at <http://ssrn.com/abstract=2208240>. Data, including Web pages and ads, archived at <http://foreverdata.org/onlineads>).
16. U.S. Commission on Civil Rights. 1970. Racism in America and how to combat it. Washington, DC.
17. WebShot Command Line Server Edition. Version 1.9.1.1; <http://www.websitescreenshots.com/>.

### LOVE IT, HATE IT? LET US KNOW

[feedback@queue.acm.org](mailto:feedback@queue.acm.org)

**LATANYA SWEENEY** ([latanya@fas.harvard.edu](mailto:latanya@fas.harvard.edu)) is professor of government and technology in residence at Harvard University. She creates and uses technology to assess and solve societal, political, and governance problems and teaches others how to do the same. She is also founder and director of the Data Privacy Lab at Harvard. She earned her Ph.D. in computer science from MIT in 2001. More information about her is available at [latanyasweeney.org](http://latanyasweeney.org).

© 2013 ACM 1542-7730/13/0300 \$10.00



# acmqueue Eventual Consistency Today: Limitations, Extensions, and Beyond

**How can applications be built on eventually consistent infrastructure given no guarantee of safety?**

Peter Bailis and Ali Ghodsi, UC Berkeley

In a July 2000 conference keynote, Eric Brewer, now VP of engineering at Google and a professor at the University of California, Berkeley, publicly postulated the CAP (consistency, availability, and partition tolerance) theorem, which would change the landscape of how distributed storage systems were architected.<sup>8</sup> Brewer's conjecture—based on his experiences building infrastructure for some of the first Internet search engines at Inktomi—states that distributed systems requiring always-on, highly available operation cannot guarantee the illusion of coherent, consistent single-system operation in the presence of network partitions, which cut communication between active servers. Brewer's conjecture proved prescient: in the following decade, with the continued rise of large-scale Internet services, distributed-system architects frequently dropped “strong” guarantees in favor of weaker models—the most notable being *eventual consistency*.

Eventual consistency provides few guarantees. Informally, it guarantees that, if no *additional* updates are made to a given data item, all reads to that item will eventually return the same value. This is a particularly weak model. At no given time can the user rule out the possibility of inconsistent behavior: the system can return *any* data and still be eventually consistent—as it might “converge” at some later point. The only guarantee is that, at some point in the future, something good will happen. Yet, despite this apparent lack of useful guarantees, scores of usable applications and profitable businesses are built on top of eventually consistent infrastructure. How?

This article begins to answer this question by describing several notable developments in the theory and practice of eventual consistency, with a focus on immediately applicable takeaways for practitioners running distributed systems in the wild. As production deployments have increasingly adopted weak consistency models such as eventual consistency, we have learned several lessons about how to reason about, program, and strengthen these weak models.

We will primarily focus on three questions and some preliminary answers:

**How eventual is eventual consistency?** If the scores of system architects advocating eventual consistency are any indication, eventual consistency seems to work “well enough” in practice. How is this possible when it provides such weak guarantees? *New prediction and measurement techniques allow system architects to quantify the behavior of real-world eventually consistent systems. When verified via measurement, these systems appear strongly consistent most of the time.*

**How should one program under eventual consistency?** How can system architects cope with the lack of guarantees provided by eventual consistency? How do they program without strong ordering guarantees? *New research enables system architects to deal with inconsistencies, either via external compensation outside of the system or by limiting themselves to data structures that avoid inconsistencies altogether.*

**Is it possible to provide stronger guarantees than eventual consistency without losing its benefits?** In addition to guaranteeing eventual consistency and high availability, what other

guarantees can be provided? *Recent results show that it's possible to achieve the benefits of eventual consistency while providing substantially stronger guarantees, including causality and several ACID (atomicity, consistency, isolation, durability) properties from traditional database systems while still remaining highly available.*

This article is *not* intended as a formal survey of the literature surrounding eventual consistency. Rather, it is a pragmatic introduction to several developments on the cutting edge of our understanding of eventually consistent systems. The goal is to provide the necessary background for understanding both *how* and *why* eventually consistent systems are programmed, are deployed, and have evolved, as well as where the systems of tomorrow are heading.

#### EVENTUAL CONSISTENCY: HISTORY AND CONCEPTS

Brewer's CAP theorem dictates that it is impossible simultaneously to achieve always-on experience (*availability*) and to ensure that users read the latest written version of a distributed database (*consistency*—as formally proven, a property known as “linearizability”<sup>11</sup>) in the presence of partial failure (*partitions*).<sup>8</sup> CAP pithily summarizes tradeoffs inherent in decades of distributed-system designs (e.g., RFC 677<sup>14</sup> from 1975) and shows that maintaining an SSI (single-system image) in a distributed system has a cost<sup>10</sup>. If two processes (or groups of processes) within a distributed system cannot communicate (are *partitioned*)—either because of a network failure or the failure of one of the components—then updates cannot be synchronously propagated to all processes without blocking. Under partitions, an SSI system cannot safely complete updates and hence is unavailable to some or all of its users. Moreover, even without partitions, a system that chooses availability over consistency enjoys benefits of low latency: if a server can safely respond to a user's request when it is partitioned from all other servers, then it can *also* respond to a user's request without contacting other servers even when it is able to do so.<sup>1</sup> (Note that you can't “sacrifice” partition tolerance!<sup>12</sup> The choice is between consistency and availability.)

As services are increasingly replicated to provide fault tolerance (ensuring that services remain online despite individual server failures) and capacity (to allow systems to scale with variable request rates), architects must face these consistency-availability and consistency-latency tradeoffs head on. In a dynamic, partitionable Internet, services requiring guaranteed low latency must often relax their expectations of data consistency.

#### EVENTUAL CONSISTENCY AS AN AVAILABLE ALTERNATIVE

Given the CAP impossibility result, distributed-database designers sought weaker consistency models that would enable both availability and high performance. While weak consistency has been studied and deployed in various forms since the 1970s,<sup>19</sup> the eventual consistency model has become prominent, particularly among emerging, highly scalable NoSQL stores.

One of the earliest definitions of eventual consistency comes from a 1988 paper describing a group communication system<sup>15</sup> not unlike a shared text editor such as Google Docs today: “...changes made to one copy eventually migrate to all. If all update activity stops, after a period of time all replicas of the database will converge to be logically equivalent: each copy of the database will contain, in a predictable order, the same documents; replicas of each document will contain the same fields.”

Under eventual consistency, all servers eventually “converge” to the same state; at some point in the future, servers are indistinguishable from one another. This eventual convergence, however,

does not provide SSI semantics. First, the “predictable order” will not necessarily correspond to an execution that could have arisen under SSI; eventual consistency does not specify which value is eventually chosen. Second, there is an unspecified window before convergence is reached, during which the system will not provide SSI semantics, but rather arbitrary values. As will be seen shortly, this promise of eventual convergence is a rather weak property. Finally, a system with SSI provides eventual consistency—the “eventuality” is immediate—but not vice versa.

Why is eventual consistency useful? Pretend you are in charge of the data infrastructure at a social network where users post new status updates that are sent to their followers’ timelines, represented by separate lists—one per user. Because of large scale and frequent server failures, the database of timelines is stored across multiple physical servers. In the event of a partition between two servers, however, you cannot deliver each update to all timelines. What should you do? Should you tell the user that he or she cannot post an update, or should you wait until the partition heals before providing a response? Both of these strategies choose consistency over availability, at the cost of user experience.

Instead, what if you propagate the update to the reachable set of followers’ timelines, return to the user, and delay delivering the update to the other followers until the partition heals? In choosing this option, you give up the guarantee that all users see the same set of updates at every point in time (and admit the possibility of timeline reordering as partitions heal), but you gain high availability and (arguably) a better user experience. Moreover, because updates are eventually delivered, all users eventually see the same timeline with all of the updates that users posted.

#### IMPLEMENTING EVENTUAL CONSISTENCY

A key benefit of eventual consistency is that it is fairly straightforward to implement. To ensure convergence, replicas must exchange information with one another about which writes they have seen. This information exchange is often called *anti-entropy*, a homage to the process of reversing entropy, or thermodynamic randomness, in a physical system.<sup>19</sup> Protocols for achieving anti-entropy take a variety of forms; one simple solution is to use an asynchronous all-to-all broadcast: when a replica receives a write to a data item, it immediately responds to the user, then, in the background, sends the write to all other replicas, which in turn update their locally stored data items. In the event of concurrent writes to a given data item, replicas deterministically choose a “winning” value, often using a simple rule such as “last writer wins” (e.g., via a clock value embedded in each write).<sup>22</sup>

Suppose you want to make a single-node database into an eventually consistent distributed database. When you get a request, you route it to any server you can contact. When a server performs a write to its local key-value store, it can send the write to all other servers in the cluster. This write-forwarding becomes the anti-entropy process. Be careful, however, when sending the write to the other servers. If you wait for other servers to respond before acknowledging the local write, then, if another server is down or partitioned from you, the write request will hang indefinitely. Instead, you should send the request in the background; anti-entropy should be an asynchronous process. Implicitly, the model for eventual consistency assumes that system partitions are eventually healed and updates are eventually propagated, or that partitioned nodes eventually die and the system ends up operating in a single partition.

The eventually consistent system has some great properties. It does not require writing difficult “corner-case” code to deal with complicated scenarios such as downed replicas or network

partitions—anti-entropy will simply stall—or writing complex code for coordination such as master election. All operations complete locally, meaning latency will be bounded. In a geo-replicated scenario, with replicas located in different data centers, you don't have to endure long-haul wide-area network latencies on the order of hundreds of milliseconds on the request fast path. The mechanism just described, returning immediately on the local write, can put data durability at risk. An intermediate point in trading between durability and availability is to return after  $W$  replicas have acknowledged the write, thus allowing the write to survive  $W-1$  replica failures. Anti-entropy can be run as often or as rarely as desired without violating any guarantees. What's not to like?

#### SAFETY AND LIVENESS

While eventual consistency is relatively easy to achieve, the current definition leaves some unfortunate holes. First, what is the eventual state of the database? A database always returning the value 42 is eventually consistent, even if 42 was never written. Amazon CTO Werner Vogels' preferred definition specifies that “eventually all accesses return the last updated value”; accordingly, the database cannot converge to an arbitrary value.<sup>23</sup> Even this new definition has another problem: what values can be returned before the eventual state of the database is reached? If replicas have not yet converged, what guarantees can be made about the data returned?

These questions stem from two kinds of properties possessed by all distributed systems: safety and liveness.<sup>2</sup> A *safety* property guarantees that “nothing bad happens;” for example, every value that is read was, at some point in time, written to the database. A *liveness* property guarantees that “something good eventually happens;” for example, all requests eventually receive a response.

The difficulty with eventual consistency is that it makes no safety guarantees—eventual consistency is purely a liveness property. Something good eventually happens—the replicas agree—but there are no guarantees with respect to what happens, and no behavior is ruled out in the meantime! For meaningful guarantees, safety and liveness properties need to be taken together: without one or the other, you can have trivial implementations that provide less-than-satisfactory results.

Virtually every other model that is stronger than eventual provides some form of safety guarantees. For almost all production systems, however, eventual consistency should be considered a bare-minimum requirement for data consistency. A system that does not guarantee replica convergence is remarkably difficult to reason about.

#### HOW EVENTUAL IS EVENTUAL CONSISTENCY?

Despite the lack of safety guarantees, eventually consistent data stores are widely deployed. Why? While eventually consistent stores don't promise safety, there is evidence that eventual consistency works well in practice. Eventual consistency is “good enough,” given its latency and availability benefits. For the many stores that offer a choice between eventual consistency and stronger consistency models, scores of practitioners advocate eventual consistency.

The behavior of eventually consistent stores can be quantified. Just because eventual consistency doesn't promise safety doesn't mean safety isn't often provided—and you can both measure and predict these properties of eventually consistent systems using a range of techniques that have recently been developed and are making their way to production stores. These techniques—which we discuss next—have surprisingly shown that eventual consistency often behaves like strong consistency in production stores.

## METRICS AND MECHANISMS

One common metric for eventual consistency is *time*: how long will it take for writes to become visible to readers? This captures the “window of consistency” measured according to the wall clock. Another metric is *versions*: how many versions old will a given read be? This information can be used to ensure that readers never go back in time, but always observe progressively newer versions of the database. While time and versions are perhaps the most intuitive metrics, there are a range of others, such as numerical drift from the “true” value of each data item and various combinations of these metrics.<sup>25</sup>

The two main kinds of mechanisms for quantifying eventual consistency are measurement and prediction. *Measurement* answers the question, “How consistent is my store under my given workload right now?”<sup>18</sup> while *prediction* answers the question, “How consistent will my store be under a given configuration and workload?”<sup>4</sup> Measurement is useful for runtime monitoring and alerts or verifying compliance with SLOs (service-level objectives). Prediction is useful for probabilistic what-if analyses such as the effect of configuration and workload changes and for dynamically tuning system behavior. Taken together, measurement and prediction form a useful toolkit.

## PROBABILISTICALLY BOUNDED STALENESS

As a brief deep dive into how to quantify eventually consistent behavior, we will discuss our experiences developing, deploying, and integrating state-of-the-art prediction techniques into Cassandra, a popular NoSQL. Probabilistically Bounded Staleness, or PBS, provides an *expectation* of recency for reads of data items.<sup>4</sup> This allows us to measure how far an eventually consistent store’s behavior deviates from that of a strongly consistent, linearizable (or regular) store. PBS enables metrics of the form: “100 milliseconds after a write completes, 99.9 percent of reads will return the most recent version,” and “85 percent of reads will return a version that is within two of the most recent.”

## BUILDING PBS

How does PBS work? Intuitively, the degree of inconsistency is determined by the rate of anti-entropy. If replicas constantly exchange their last-written writes, then the window of inconsistency should be bounded by the network delay and local processing delay at each node. If replicas delay anti-entropy (possibly to save bandwidth or processing time), then this delay is added to the window of inconsistency; many systems (Amazon’s Dynamo, for example) offer settings in the replication protocol to control these delays. Given the anti-entropy protocol, then—given the configured anti-entropy rate, the network delay, and local processing delay—you can calculate the expected consistency. In Cassandra, we piggyback timing information on top of the write distribution protocol (the primary source of anti-entropy) and maintain a running sample. When a user wants to know the effect of a given replication configuration, we use the collected sample in a Monte Carlo simulation of the protocol to return an expected value for the consistency of the data store, which closely matches consistency measurements on our Cassandra clusters at Berkeley.

## PBS IN THE WILD

Using our PBS consistency prediction tool, and with the help of several friends at LinkedIn and Yammer, we quantified the consistency of three eventually consistent stores running in production. PBS models predicted that LinkedIn’s data stores returned consistent data 99.9 percent of the



time within 13.6 ms, and on SSDs (solid-state drives) within 1.63 ms. These eventually consistent configurations were 16.5 percent and 59.5 percent faster than their strongly consistent counterparts at the 99.9<sup>th</sup> percentile. Yammer's data stores experienced a 99.9 percent inconsistency window of 202 ms at 81.1 percent latency reduction. The results confirmed the anecdotal evidence: eventually consistent stores are often faster than their strongly consistent counterparts, and they are frequently consistent within tens or hundreds of milliseconds.

In order to make consistency prediction more accessible, with the help of the Cassandra community, we recently released support for PBS predictions in Cassandra 1.2.0. Cassandra users can now run predictions on their own production clusters to tune their consistency parameters and perform what-if analyses for normal-case, failure-free operation. For example, to explore the effect of adding SSDs to a set of servers, users can adjust the expected distribution of read and write speeds on the local node. These predictions are inexpensive; a JavaScript-based demonstration we created<sup>4</sup> completes tens of thousands of trials in less than a second.

Of course, prediction is not without faults: predictions are only as good as the underlying model and input data. As statistician George E.P. Box famously stated, "All models are wrong, but some are useful." Failure to account for an important aspect of the system or anti-entropy protocol may lead to inaccurate predictions. Similarly, prediction works by assuming that past behavior is correlated with future behavior. If environmental conditions change, predictions may be of limited accuracy. These issues are fundamental to the problem at hand, and they are a reminder that prediction is best paired with measurement to ensure accuracy.

#### EVENTUAL CONSISTENCY IS OFTEN STRONGLY CONSISTENT

In addition to PBS, several recent projects have verified the consistency of real-world eventually consistent stores. One study found that Amazon SimpleDB's inconsistency window for eventually consistent reads was almost always less than 500 ms,<sup>24</sup> while another study found that Amazon S3's inconsistency window lasted up to 12 seconds.<sup>7</sup> Other recent work shows results similar to those presented for PBS, with Cassandra closing its inconsistency window within around 200 ms.<sup>18</sup>

These results confirm the anecdotal evidence that eventual consistency is often "good enough" by providing quantitative metrics for system behavior. As techniques such as PBS and consistency measurement continue to make their way into more production infrastructure, reasoning about the behavior of eventual consistency across deployments, failures, and system configurations will be increasingly straightforward.

#### PROGRAMMING EVENTUAL CONSISTENCY

While users can verify and predict the consistency behavior of eventually consistent systems, these techniques do not provide absolute guarantees against safety violations. What if an application requires that safety is always respected? There is a growing body of knowledge about how to program and reason about eventually consistent stores.

#### COMPENSATION, COSTS, AND BENEFITS

Programming around consistency anomalies is similar to speculation: you don't know what the latest value of a given data item is, but you can proceed as if the value presented is the latest. When you've guessed wrong, you have to compensate for any incorrect actions taken in the interim.

In effect, compensation is a way to achieve safety retroactively—to restore guarantees to users.<sup>13</sup> Compensation ensures that mistakes are eventually corrected but does not guarantee that no mistakes are made.

As an example of speculation and compensation, consider running an ATM machine.<sup>8,13</sup> Without strong consistency, two users might simultaneously withdraw money from an account and end up with more money than the account ever held. Would a bank ever want this behavior? In practice, yes. An ATM's ability to dispense money (availability) outweighs the cost of temporary inconsistency in the event that an ATM is partitioned from the master bank branch's servers. In the event of overdrawing an account, banks have a well-defined system of external compensating actions: for example, overdraft fees charged to the user. Banking software is often used to illustrate the need for strong consistency, but in practice the socio-technical system of the bank can deal with data inconsistency just as well as with other errors such as data-entry mistakes.

An application designer deciding whether to use eventual consistency faces a choice. In effect, the designer needs to weigh the benefit of weak consistency  $B$  (in terms of high availability or low latency) against the cost  $C$  of each inconsistency anomaly multiplied by the rate of anomalies  $R$ :

maximize  $B - CR$

This decision is, by necessity, application- and deployment-specific. The cost of anomalies is determined by the cost of compensation: too many overdrafts might cause customers to leave a bank, while too-slow propagation of status updates might cause users to leave a social network. The rate of anomalies—as seen before—depends on the system architecture, configuration, and deployment. Similarly, the benefit of weak consistency is itself possibly a compound term composed of factors such as the incidence of communication failures and communication latency.

Second, application designers actually have to design for compensation. Writing corner-case compensation code is nontrivial. Determining the correct business application logic to handle each type of consistency anomaly is a difficult task. Carefully reasoning about each possible sequence of anomalies and the correct “apologies” to make to the user for each can become more onerous than designing a solution for strong consistency. In general, when the cost of inconsistency is high, with tangible monetary consequences (e.g., ATMs), compensation is more likely to be well thought out. Additionally, depending on the application, it is possible that some compensation protocols already exist. For example, even if a database is perfectly consistent, a forklift may run over a pallet of inventory in a warehouse or packages may be lost in transit.<sup>13</sup>

For some applications, however, the rate of anomalies may be low enough or the cost of inconsistency may be small enough that the application designer may choose to forgo including compensation entirely. If the chance of inconsistency is sufficiently low, users may experience anomalies in only a small number of cases. Anecdotally, many online services such as social networking largely operate with weakly consistent configurations: if a user's status update takes seconds or even minutes to propagate to followers, they are unlikely to notice or even care. The complexities of operating a strongly consistent service at this scale may outweigh the benefit of, say, preventing an off-by-one error in Justin Bieber's follower count on Twitter.

## COMPENSATION BY DESIGN

Compensation is error-prone and laborious, and it exposes the programmer (and sometimes the application) to the effects of replication. What if you could program without it? Recent research has provided “compensation-free” programming for many eventually consistent applications.

The formal underpinnings of eventually consistent programs that are consistent by design are captured by the CALM theorem, indicating which programs are safe under eventual consistency and also (conservatively) which aren't.<sup>3</sup> Formally, CALM means *consistency as logical monotonicity*; informally, it means that programs that are monotonic, or compute an ever-growing set of facts (by, e.g., receiving new messages or performing operations on behalf of a client) and do not ever “retract” facts that they emit (i.e., the basis for decisions the program has already made doesn't change), can always be safely run on an eventually consistent store. (Full disclosure: CALM was developed by our colleagues at UC Berkeley). Accordingly, CALM tells programmers which operations and programs can guarantee safety when used in an eventually consistent system. Any code that fails CALM tests is a candidate for stronger coordination mechanisms.

As a concrete example of this logical monotonicity, consider building a database for queries on stock trades. Once completed, trades cannot change, so any answers that are based solely on the immutable *historical* data will remain true. However, if your database keeps track of the value of the *latest* trade, then new information—such new stock prices—might retract old information, as new stock prices overwrite the latest ones in the database. Without coordination between replica copies, the second database might return inconsistent data.

By analyzing programs for monotonicity, you can “bless” monotonic programs as “safe” under eventual consistency and encourage the use of coordination protocols (i.e., strong consistency) in the presence of non-monotonicity. As a general rule, operations such as initializing variables, accumulating set members, and testing a threshold condition are monotonic. In contrast, operations such as variable overwrites, set deletion, counter resets, and negation (e.g., “there does not exist a trade such that...”) are generally not logically monotonic.

CALM captures a wide space of design patterns sometimes referred to as ACID 2.0 (associativity, commutativity, idempotence, and distributed)<sup>13</sup>. *Associativity* means that you can apply a function in any order:

$$f(a, f(b, c)) = f(f(a,b),c)$$

*Commutativity* means that a function's arguments are order-insensitive:

$$f(a,b) = f(b,a)$$

Commutative and associative programs are order-insensitive and can tolerate message re-ordering, as in eventual consistency. *Idempotence* means you can call a function on the same input any number of times and get the same result:

$$f(f(x))=f(x) \text{ (e.g., } \max(42, \max(42, 42)) = 42)$$

Idempotence allows the use of at-least-once message delivery, instead of at-most-once delivery (which is more expensive to guarantee). *Distributed* is primarily a placeholder for *D* in the acronym (!) but symbolizes the fact that ACID 2.0 is all about distributed systems. Carefully applying these design patterns can achieve logical monotonicity.

Recent work on CRDTs (commutative, replicated data types) embodies CALM and ACID 2.0 principles within a variety of standard data types, providing provably eventually consistent data structures including sets, graphs, and sequences.<sup>20</sup> Any program that correctly uses these predefined, well-specified data structures is guaranteed to never produce any safety violations.

To understand CRDTs, consider building an increment-only counter that is replicated on two servers. We might implement the increment operation by first reading the counter's value on one replica, incrementing the value by one, and writing the new value back on every replica. If the counter is initially at 0 and two different users simultaneously initiate increment operations on separate servers, both users may read 0 and then distribute the value 1 to the replicas; the counter ends up with a value of 1 instead of the correct value of 2. Instead, we can use a G-counter CRDT, which relies on the fact that *increment* is a commutative operation—it doesn't matter in what order the two *increment* operations are applied, as long as they are both eventually applied at all sites. With a G-counter, the current counter status is represented as the count of distinct *increment* invocations, similar to how counting is introduced at the grade-school level: by making a tally mark for every increment then summing the total. In our example, instead of reading and writing counter *values*, each invocation distributes an increment *operation*. All replicas end up with two increment operations, which sum to the correct value of 2. This works because the replicas understand the semantics of increment operations instead of providing general-purpose read/write operations, which are not commutative.

A key property of these advances is that they separate data store and application-level consistency concerns. While the underlying store may return inconsistent data at the level of reads and writes, CALM, ACID 2.0, and CRDT appeal to *higher-level* consistency criteria, typically in the form of application-level invariants that the application maintains. Instead of requiring that every read and write to and from the data store is strongly consistent, the application simply has to ensure a semantic guarantee (such as “the counter is strictly increasing”)—granting considerable leeway in how reads and writes are processed. This distinction between application-level and read/write consistency is often ambiguous and poorly defined (for example, what does database ACID “consistency” have to do with “strong consistency”?). Fortunately, by identifying a large class of programs and data types that are tolerant of weak consistency, programmers can enjoy “strong” application consistency, while reaping the benefits of “weak” distributed read/write consistency.

Taken together, the CALM theorem and CRDTs make a powerful toolkit for achieving “consistency without concurrency control,” which is making its way into real-world systems. Our team's work on the Bloom language<sup>3</sup> embodies CALM principles. Bloom encourages the use of order-insensitive disorderly programming, which is key to architecting eventually consistent systems. Some of our recent work focuses on building custom eventually consistent data types whose correctness is grounded in formal mathematical lattice theory. Concurrently, several open source projects such as Statebox<sup>21</sup> provide CRDT-like primitives as client-side extensions to eventually consistent stores, while one eventually consistent store—Riak—recently announced alpha support for CRDTs as a first-class server-side primitive.<sup>9</sup>

## STRONGER THAN EVENTUAL

While compensating actions and CALM/CRDTs provide a way around eventual consistency, they have shortcomings of their own. The former requires dealing with inconsistencies outside the system and the latter limits the operations that an application writer can employ. However, it turns out that it is possible to provide even stronger guarantees than eventual consistency—albeit weaker than SSI—for general-purpose operations while still providing availability.

The CAP theorem dictates that strong consistency (SSI) and availability are unachievable in the

presence of partitions. But how weak does the consistency model have to be in order for it to be available? Clearly, eventual consistency, which simply provides a liveness guarantee, is available. Is it possible to strengthen eventual consistency by adding safety guarantees to it without losing its benefits?

#### PUSHING THE LIMITS

A recent technical report from the University of Texas at Austin claims that no consistency model stronger than causal consistency is available in the presence of partitions.<sup>17</sup> Causal consistency guarantees that each process's writes are seen in order, that writes follow reads (if a user reads a value  $A=5$  and then writes  $B=10$ , then another user cannot read  $B=10$  and subsequently read an older value of  $A$  than 5), and that transitive data dependencies hold. This causal consistency is useful in making sure, for example, that comment threads are seen in the correct order, without dangling replies, and that users' privacy settings are applied to the appropriate data. The UT Austin report demonstrates that it is not possible to have a stronger model than causal consistency (that accepts fewer outcomes) without either violating high availability or giving up the assurance that, if two servers communicate, they will agree on the same set of values for their data items. While many other available models are neither stronger nor weaker than causal consistency, this impossibility result is useful because it places an upper bound on a very familiar consistency model.

Especially in light of this result, it is worth noting that several new data storage designs provide causal consistency. The COPS and Eiger systems<sup>16</sup> developed by a team from Princeton, CMU, and Intel Research provide causal consistency without incurring high latencies across geographically distant datacenters or the loss of availability in the event of datacenter failures. These systems perform particularly well, at a near-negligible cost to performance when compared to eventual consistency; Eiger, which was prototyped within the Cassandra system, incurs less than 7% overhead for one of Facebook's workloads. In our recent work, we demonstrated how existing data stores that are already deployed in production but provide eventual consistency can be augmented with causality as an added safety guarantee.<sup>6</sup> Causality can be *bolted-on* without compromising high availability, enabling system designs in which safety and liveness are cleanly decomposed into separate architectural layers.

In addition to causality, we can consider the relationship between ACID transactions and the CAP theorem. While it's impossible to provide the gold standard of ACID isolation—serializability, or SSI—it turns out that many ACID databases provide a weaker form of isolation, such as read committed, often by default and, in some cases, as the maximum offered. Some of our recent results show that many of these weaker models *can* be implemented in a distributed environment while providing high availability.<sup>5</sup> Current databases providing these weak isolation models are unavailable, but this is only because they have been implemented with unavailable algorithms.

We—and several others—are developing transactional algorithms that show this need not be the case. By rethinking the concurrency-control mechanisms and re-architecting distributed databases from the ground up, we can provide safety guarantees in the form of transactional atomicity, ANSI SQL Read Committed and Repeatable Read, and causality between transactions—matching many existing ACID databases—without violating high availability. This is somewhat surprising, as many in the past have assumed that, in a highly available system, arbitrary multi-object transactions are out of the question.



## RECOGNIZING THE LIMITS

While these results push the limits of what is achievable with high availability, there are several properties that a weakly consistent system will never be able to provide; there is a fundamental cost to remaining highly available (and providing guaranteed low latency). The CAP theorem states that staleness guarantees are impossible in a highly available system. Reads that specify a constraint on data recency (e.g., “give me the latest value” or “give me the latest value as of 10 minutes ago”) are not generally available in the presence of long-lasting network partitions. Similarly, we cannot maintain arbitrary global correctness constraints over sets of data items such as uniqueness requirements (e.g., “create bank account with ID 50 if the account does not exist”) and, in certain cases (e.g., arbitrary reads and writes), even correctness constraints on individual data items are not achievable (e.g., “the bank account balance should be non-negative”). These challenges are an inherent cost of choosing weak consistency—whether eventual or a stronger but still “weak” model.

## CONCLUSIONS

By simplifying the design and operation of distributed services, eventual consistency improves availability and performance at the cost of semantic guarantees to applications. While eventual consistency is a particularly weak property, eventually consistent stores often deliver consistent data, and new techniques for measurement and prediction grant us insight into the behavior of eventually consistent stores. Concurrently, new research and prototypes for building eventually consistent data types and programs are easing the burden of reasoning about disorder in distributed systems. These techniques, coupled with new results that push the boundaries of highly available systems—including causality and transactions—make a strong case for the continued adoption of weakly consistent systems. While eventual consistency and its weakly consistent cousins are not perfect for every task, their performance and availability will likely continue to accrue admirers and advocates in the future.

## ACKNOWLEDGMENTS

The authors would like to thank Peter Alvaro, Carlos Baquero, Neil Conway, Alan Fekete, Joe Hellerstein, Marc Shapiro, and Ion Stoica for feedback on earlier drafts of this article.

This work was supported by gifts from Google, SAP, Amazon Web Services, Blue Goji, Cloudera, Ericsson, General Electric, Hewlett Packard, Huawei, IBM, Intel, MarkLogic, Microsoft, NEC Labs, NetApp, NTT Multimedia Communications Laboratories, Oracle, Quanta, Splunk, and VMware. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant DGE 1106400, National Science Foundation Grants IIS-0713661, CNS-0722077, and IIS-0803690, the Air Force Office of Scientific Research Grant FA95500810352, and DARPA contract FA865011C7136.

## REFERENCES

1. Abadi, D. 2012. Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. *IEEE Computer* (February).
2. Alpern, B., Schneider, F.B. 1985. Defining liveness. *Information Processing Letters* 21 (October).
3. Alvaro, P., Conway, N., Hellerstein, J., Marczak, W. 2011. Consistency analysis in Bloom: a CALM and collected approach. *CIDR (Conference on Innovative Data Systems Research)*.

4. Bailis, P., Venkataraman, S., Franklin, M., Hellerstein, J., Stoica, I. 2012. Probabilistically bounded staleness for practical partial quorums. VLDB (Very Large Databases). (Demo from text: <http://pbs.cs.berkeley.edu/#demo>)
5. Bailis, P., Fekete, A., Ghodsi, A., Hellerstein, J., Stoica, I. 2013. HAT, not CAP: highly available transactions. arXiv:1302.0309 [cs.DB] (February).
6. Bailis, P., Ghodsi, A., Hellerstein, J., Stoica, I. 2013. Bolt-on causal consistency. ACM SIGMOD.
7. Bermbach, D., Tai, S. 2011. Eventual consistency: how soon is eventual? An evaluation of Amazon S3's consistency behavior. MW4SOC (Workshop on Middleware for Service-oriented Computing).
8. Brewer, E. 2012. CAP twelve years later: how the "rules" have changed. IEEE Computer (February).
9. Brown, R., Cribbs, S. 2012. Data structures in Riak; <https://speakerdeck.com/basho/data-structures-in-riak>. RICON Conference.
10. Davidson, S., Garcia-Molina, H., Skeen, D. 1985. Consistency in a partitioned network: a survey. ACM Computing Surveys Volume 17, Issue 3.
11. Gilbert, S., Lynch, N. 2002. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News Volume 33, Issue 2 (June).
12. Hale, C. 2010. You can't sacrifice partition tolerance. <http://codahale.com/you-cant-sacrifice-partition-tolerance/>
13. Helland, P., Campbell, D. 2009. Building on quicksand. CIDR (Conference on Innovative Data Systems Research).
14. Johnson, P. R., Thomas, R. H. 1975. Maintenance of duplicate databases; RFC 677; <http://www.faqs.org/rfcs/rfc677.html>.
15. Kawell Jr., L., Beckhardt, S., Halvorsen, T., Ozzie, R., Greif, I. 1988. Replicated document management in a group communication system. *Proceedings of the 1988 ACM Conference on Computer-supported Cooperative Work*: 395; <http://dl.acm.org/citation.cfm?id=1024798>.
16. Lloyd, W., Freedman, M., Kaminsky, M., Andersen, D. 2013. Stronger semantics for low-latency geo-replicated storage. NSDI (Networked Systems Design and Implementation).
17. Mahajan, P., Alvisi, L., Dahlin, M. 2011. Consistency, availability, convergence. University of Texas at Austin TR-11-22 (May).
18. Rahman, M., Golab, W., AuYoung, A., Keeton, K., Wylie, J. 2012. Toward a principled framework for benchmarking consistency. HotDep (Workshop on Hot Topics in System Dependability).
19. Saito, Y., Shapiro, M. 2005. Optimistic replication. ACM Computing Surveys Volume 37 Number 1 (March). <http://dl.acm.org/citation.cfm?id=1057980>
20. Shapiro, M., Preguiça, N., Baquero, C., Zawirski, M. 2011. A comprehensive study of convergent and commutative replicated data types. INRIA Technical Report RR-7506 (January).
21. Statebox; <https://github.com/mochi/statebox>.
22. Terry, D., Theimer, M., Petersen, K., Demers, A., Spreitzer, M. Hauser, C. 1995. Managing update conflicts in Bayou, a weakly connected replicated storage system. SOSP (Symposium on Operating Systems Principles).
23. Vogels, W. Eventually consistent. 2008. ACM Queue.
24. Wada, H., Fekete, A., Zhao, L., Lee, K., A. Liu, A. 2011. Data consistency and the tradeoffs in commercial cloud storage: the consumers' perspective. CIDR (Conference on Innovative Data Systems Research).

25. Yu, H., Vahdat, A. 2002. Design and evaluation of a conit-based continuous consistency model for replicated services. ACM TOCS (Transactions on Computer Systems).

#### RECOMMENDED READING

##### Compensation and Stronger Models

Shapiro, M., Preguiça, N., Baquero, C., Zawirski, M. 2011. A comprehensive study of convergent and commutative replicated data types. INRIA Technical Report RR-7506 (January). <http://hal.upmc.fr/docs/00/55/55/88/PDF/techreport.pdf>

Terry, D. 2011. Replicated data consistency explained through baseball. Microsoft Research Technical Report MSR-TR-2011-137 (October). <http://research.microsoft.com/apps/pubs/default.aspx?id=157411>

Saito, Y., Shapiro, M. 2005. Optimistic Replication. ACM Computing Surveys Volume 37 Number 1 (March). <http://dl.acm.org/citation.cfm?id=1057980>

Helland, P., Campbell, D. 2009. Building on quicksand. CIDR (Conference on Innovative Data Systems Research). [http://www-db.cs.wisc.edu/cidr/cidr2009/Paper\\_133.pdf](http://www-db.cs.wisc.edu/cidr/cidr2009/Paper_133.pdf)

##### CAP and Latency-Consistency Background

Abadi, D.J. 2012. Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. *IEEE Computer* (February). <http://cs-www.cs.yale.edu/homes/dna/papers/abadi-pacelc.pdf>

Brewer, E. 2012. CAP twelve years later: how the “rules” have changed. *IEEE Computer* (February). <http://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed>

*Portions of this piece (in particular, the safety and liveness discussion) originally appeared at <http://bailis.org/blog>.*

##### LOVE IT, HATE IT? LET US KNOW

[feedback@queue.acm.org](mailto:feedback@queue.acm.org)

**PETER BAILIS** is a graduate student of Computer Science in the AMPLab and BOOM projects at UC Berkeley, where he works closely with Ali Ghodsi, Joe Hellerstein, and Ion Stoica. He currently studies distributed systems and databases, with a particular focus on distributed consistency models. Peter received his A.B. from Harvard College and is the recipient of the NSF Graduate Research Fellowship and the Berkeley Fellowship for Graduate Study. Peter blogs regularly at <http://bailis.org/blog> and tweets as @pbailis.

**ALI GHODSI** is an Assistant Professor at KTH/Royal Institute of Technology in Sweden and a Visiting Researcher at UC Berkeley since 2009. His general interests are in the broader areas of distributed systems and networking. He received his PhD in 2006 from KTH/Royal Institute of Technology in the area of Distributed Computing. He can be reached at [alig@cs.berkeley.edu](mailto:alig@cs.berkeley.edu).

# acmqueue

## A File System All Its Own

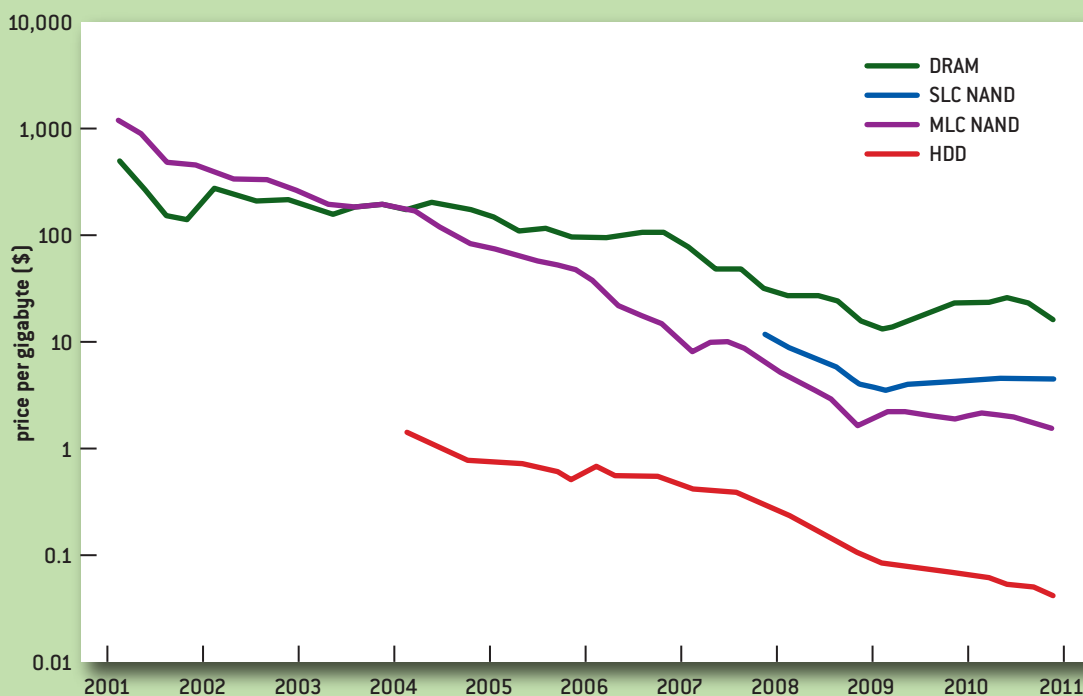
Flash memory has come a long way. Now it's time for software to catch up.

Adam H. Leventhal

In the past five years, flash memory has progressed from a promising accelerator,<sup>7</sup> whose place in the data center was still uncertain, to an established enterprise component for storing performance-critical data<sup>4,9</sup>. It's rise to prominence followed its proliferation in the consumer world and the volume economics that followed (see figure 1). With SSDs (solid-state devices), flash arrived in a form optimized for compatibility—just replace a hard drive with an SSD for radically better performance. But the properties of the NAND flash memory used by SSDs differ significantly from those of the magnetic media in the hard drives they often displace.<sup>2</sup> While SSDs have become more pervasive in a variety of uses, the industry has only just started to design storage systems that embrace the nuances of flash memory. As it escapes the confines of compatibility, significant improvements in performance, reliability, and cost are possible.

### FIGURE 1

Price Trends in the Storage Hierarchy



Source: Objective Analysis

The native operations of NAND flash memory are quite different from those required of a traditional block device. The FTL (flash translation layer), as the name suggests, translates the block-device commands into operations on flash memory. This translation is by no means trivial; both the granularity and the fundamental operations differ. SSD controllers compete in subspecialties such as garbage collection, write amplification, wear leveling, and error correction.<sup>2</sup> The algorithms used by modern SSDs are growing increasingly sophisticated despite the seemingly simple block-read and block-write operations that they must support. A very common use of a block device is to host a file system. File systems, of course, perform their own type of translation: from file creations, opens, reads, and writes within a directory hierarchy to block reads and writes. There's nothing innate about file-system operations that makes them well served by the block interface; it's just the dominant standard for persistent storage, and it has existed for decades.

Layering the file system translation on top of the flash translation is inefficient and impedes performance. Sophisticated applications such as databases have long circumvented the file system—again, layers upon layers—to attain optimal performance. The information lost between abstraction layers impedes performance, longevity, and capacity. A file system may “know” that a file is being copied, but the FTL sees each copied block as discrete and unique. File systems also optimize for the physical realities of a spinning disk, but placing data on the sectors that spin the fastest doesn't make sense when they don't spin at all. Volume managers, software that presents collections of disks as a block device, led to similar inefficiencies in disk-based storage, obscuring information from the file system.

Modern file systems such as WAFL (Write Anywhere File Layout)<sup>5</sup> ZFS, and Btrfs (B-tree file system)<sup>1</sup> integrated the responsibilities previously assigned to volume managers and reorganized the layers of abstraction. The resulting systems were more efficient and easier to manage. Poorly optimized software mattered when operations were measured in milliseconds; it matters much more on flash devices whose operations are measured in microseconds. To take full advantage of flash, users need software expressly designed for the native operations and capabilities of NAND flash.

## THE STATE OF SSDS

For many years SSDs were almost exclusively built to seamlessly replace hard drives; they not only supported the same block-device interface, but also had the same form factor (e.g., a 2.5- or 3.5-inch hard drive) and communicated using the same protocols (e.g., SATA, SAS, or FC). This is a bit like connecting an iPod to a car stereo using a tape adapter; now it seems that 30-pin iPod connectors are more common in new cars than tape decks are. Recently SSDs have started to break away from the old constraints on compatibility: some laptops now use a custom form-factor SSD for compactness, and many vendors produce PCI-attached SSDs for lower latency.

The majority of SSDs still emulate the block interface of hard drives: reading and writing an arbitrary series of sectors (512-byte or 4-KB regions). The native operations of NAND flash memory are different enough to create some substantial challenges. Reads and writes happen at the granularity of a page (usually around 8 KB) with the significant caveats that writes can occur only to erased pages, and pages are erased exclusively in blocks of 32-64 (256-512 KB). While a detailed description of how an FTL presents a block interface from flash primitives is beyond the scope of this article, it's easy to get a sense of its complexity. Consider the case of a block in which all pages have been written, and the device receives an operation to logically overwrite the contents of one page.



The FTL could copy the block into memory, modify the page, erase the block, and rewrite it in its entirety, but this would be very slow—slower even than a hard drive! In addition, each write or erase operation wears out NAND flash. Chips are rated for a certain number of such operations—anywhere from 500-50,000 cycles today depending on the type and quality, and those numbers are shrinking as the chips themselves shrink. A naive approach to block management would quickly wear down the media; and to compound the problem, a frequently overwritten region would wear out before other regions. For these reasons, FTLs use an indirection layer that allows data to be written at arbitrary locations and implements wear leveling, the process of distributing writes uniformly across the media.<sup>2</sup>

#### BRIDGING THE GAP

The algorithms that make up an FTL are highly complex but no more than those of a modern file system. Indeed, the FTL and the file system have much in common. Both track allocated versus free regions, both implement a logical-to-physical mapping, and both translate one operation set to another. Newer FTLs even include facilities such as compression and deduplication—still marquee features for modern file systems. FTLs and file systems are usually built in isolation. The idea of a dramatic integration and reorganization of the responsibilities of the FTL and file system represents a classic conundrum: who will write software for nonexistent hardware, and who will build hardware to enable heretofore-unwritten software?

Most SSD vendors are focused on a volume market where requiring a new file system on the host would be an impediment rather than an advantage. SSD vendors could enable the broader file-system developer community by providing different interfaces or opening up their firmware, but again—and without an obvious and compelling file system—there's little incentive. The exception was Indilinx's participation in the OpenSSD<sup>10</sup> project, but the primary focus was FTL development and experimentation within conventional bounds. OpenSSD became effectively defunct when OCZ acquired Indilinx. There seems to be no momentum and only vague incentive for vendors to give developers the level of visibility and control that they most want. Mainstream efforts to build flash awareness into file systems have led to more modest modifications to the interface between file system and SSD.

The most publicized interface between the file system and SSD is the ATA TRIM command or its counterpart, the SCSI UNMAP command. TRIM and UNMAP convey the same meaning to a device: the given region is no longer in use. One of the challenges with an FTL is efficient space management; and the more space that's available, the easier it is to perform that task. As free space is exhausted, FTLs have less latitude to migrate data, and they need to keep data in an increasingly compact form; with lots of free space, FTLs can be far sloppier.

For both performance and redundancy, almost all SSDs “overprovision.” They include more flash memory capacity than the advertised capacity of the SSD by anywhere from 10 to 100 percent. File systems have the notion of allocated and free blocks, but there isn't a means—or a reason—to communicate that information to a hard drive. To let SSDs reap the benefits of free storage, modern file systems use the TRIM or UNMAP commands to indicate that logical regions are no longer in use. Some SSDs—particularly those designed for the consumer market—greatly benefit from file systems that support TRIM and UNMAP. Of course, for a file system whose steady state is close to full, TRIM and UNMAP have very little impact because there aren't many free blocks.

## INCREMENTAL REVOLUTION

While many companies participate in incremental improvements, the most likely candidates to create a flash-optimized file system are those that build both SSDs and software that runs on the host. The most popularized example thus far is DirectFS<sup>6</sup> from FusionIO. Here, the flash storage provides more expressive operations for the file system. Rather than solely using the legacy block interface, DirectFS interacts with a virtualized flash storage layer. That layer manages the flash media much like a traditional FTL but offers greater visibility and an expanded set of operations to the file system above it.

DirectFS achieves significant performance improvements not by supplanting intelligence in the hardware controller, but by reorganizing responsibilities between the file system and flash controller. For example, FusionIO has proposed extensions to the SCSI standard that perform scattered reads and writes atomically.<sup>3</sup> These are easily supported by the FTL, but dramatically simplify the logic required in a file system to ensure metadata consistency in the face of a power failure. DirectFS also relies on storage that provides a “sparse address space”, which effectively transfers allocation and block mapping responsibilities from the file system to the FTL, a task the FTL already must do. A 2010 article by William Josephson et al. states that “novel layers of abstraction specifically for flash memory can yield substantial benefits in software simplicity and system performance.”<sup>6</sup>

As with TRIM, incrementally adding expressiveness and functionality to the existing storage interfaces allows file systems to take advantage of new facilities on devices that provide them. Storage system designers can choose whether to require devices that provide those interfaces or to implement a work-alike facility that they disable when it's not needed. Device vendors can decide whether supporting a richer interface represents a sufficient competitive advantage. Though this approach may never lead to an optimal state, it may allow the industry to navigate monotonically to a sufficient local maximum.

## THE CHICKEN AND THE EGG

There are still other ways to construct a storage system around flash. A more radical approach is to go further than DirectFS, assigning additional high-level responsibilities to the file system such as block management, wear leveling, read-disturb awareness, and error correction. This would allow for a complete reorganization of the software abstractions in the storage system, ensuring sufficient information for proper optimization where today's layers must cope with suboptimal information and communication. Again, this approach today requires a vendor that can assert broad control over the whole system—from the file system to the interface, controller, and flash media. It is certainly tenable for closed proprietary systems—indeed, several vendors are pursuing this approach—but for it to gain traction as a new open standard would be difficult.

## SSD ALCHEMY

The SSDs that exist today for the volume market are cheap and fast, but they exhibit performance that's inconsistent and reliability that's insufficient. Higher-level software designed with full awareness of those shortcomings could turn that commodity iron into gold. Without redesigning part or all of the I/O interface, those same SSDs could form the basis of a high-performing and highly reliable storage system.

Rather than designing a file system around the properties of NAND flash, this approach would

treat the commodity SSDs themselves as the elementary unit of raw storage. NAND flash memory already has complicated intrinsic properties; the emergent properties of an SSD are even more obscure and varied. A common pathology with SSDs, for example, is variable performance when servicing concurrent or interleaved read and write operations. Understanding these pathologies sufficiently and creating higher-level software to accommodate them would represent the flash version of an existential software parable: enterprise quality from commodity components. It's a phenomenon that the storage world has seen before with disks; software such as ZFS from Sun has produced fast, reliable systems from cheap components.

The only easy part of this transmutation is finding the base material. Building such a software system given a single, unchanging SSD would already be complicated; doing it amid the changing diversity of the SSD market further complicates the task. The properties of flash differ between types and fabrication processes, but change happens at the rate of hardware evolution. SSDs change not only to accommodate the underlying media and controller hardware, but also at the speed of software, fixing bugs and improving algorithms. Still, some vendors are pursuing this approach<sup>11</sup> because, while it is more complex than designing for purpose-built hardware, it has the potential to produce superlative systems that ride the economic curve of volume SSDs.

#### NEXT FOR FLASH

The life span of flash as a relevant technology is a topic of vigorous debate. The cost of flash memory has yet to catch up with that of hard disk drives, but prices per gigabyte are approaching those of HDDs from less than a decade ago, as shown in figure 1. While flash has ridden its price and density trends to a position of relevance, some experts anticipate fast-approaching limits to the physics of scaling NAND flash memory. Others foresee several decades of flash innovation. Whether it is flash or some other technology, nonvolatile solid-state memory will be a permanent part of the storage hierarchy, having filled the yawning gap between hard-drive and CPU speeds.<sup>8</sup>

The next evolutionary stage should see file systems designed explicitly for the properties of solid-state media rather than relying on an intermediate layer to translate. The various approaches are each imperfect. Incremental changes to the storage interface may never reach the true acme. Creating a new interface for flash might be untenable in the market. Treating SSDs as the atomic unit of storage may be just another half-measure, and a technically difficult one at that.

Some companies today are betting on the relevance of flash at least in the near term—some working within the confines of today's devices, others building, augmenting, or replacing the existing interfaces. The performance of flash memory has whetted the computer industry's appetite for faster and cheaper persistent storage. The experimentation phase is long over; it's time to build software for flash memory and embrace the specialization needed to realize its full potential.

#### LOVE IT, HATE IT? LET US KNOW

[feedback@queue.acm.org](mailto:feedback@queue.acm.org)

**ADAM H. LEVENTHAL** is the CTO at Delphix, a database virtualization company. Previously he served as Lead Flash Engineer for Sun and then Oracle where he designed flash integration in the ZFS Storage Appliance, Exadata, and other products. For over a decade, Adam has been involved in storage system design at Sun, Oracle, and now Delphix.

## REFERENCES

1. Btrfs wiki. [https://btrfs.wiki.kernel.org/index.php/Main\\_Page](https://btrfs.wiki.kernel.org/index.php/Main_Page)
2. Cornwell, M. 2012. Anatomy of a solid-state drive. *ACM Queue* 10(10); <http://queue.acm.org/detail.cfm?id=2385276>
3. Elliott, R., Batwara, A. 2012. Notes to T10 Technical Committee. 11-229r4 SBC-4 SPC-5 Atomic writes and reads <http://www.t10.org/cgi-bin/ac.pl?t=d&f=11-229r4.pdf>; 12-086r2 SBC-4 SPC-5 Scattered writes, optionally atomic <http://www.t10.org/cgi-bin/ac.pl?t=d&f=12-086r2.pdf>; 12-087r2 SBC-4 SPC-5 Gathered reads - optionally atomic <http://www.t10.org/cgi-bin/ac.pl?t=d&f=12-087r2.pdf>
4. Gray, J., Fitzgerald, B. 2008. Flash disk opportunity for server applications. *ACM Queue* 06(04); <http://queue.acm.org/detail.cfm?id=1413261>
5. Hitz, D., Lau, J.; Malcolm, M. 1994. File system design for an NFS file server appliance. WTEC'94 USENIX Winter 1994 Technical Conference: 19-19. <http://dl.acm.org/citation.cfm?id=1267093>
6. Josephson, W. K., Bongo, L. A., Li, K., Flynn, D. 2010. DFS: A file system for virtualized flash storage. *ACM Transactions on Storage (TOS)*; 6(3). <http://dl.acm.org/citation.cfm?id=1837922>
7. Leventhal, Adam. 2008. Flash storage today. *ACM Queue* 6(4); <http://queue.acm.org/detail.cfm?id=1413262>
8. Leventhal, Adam. 2009. Triple-parity RAID and beyond. *ACM Queue* 7(11); <http://queue.acm.org/detail.cfm?id=1670144>
9. Moshayedi, M., Wilkison, P. 2008. Enterprise SSDs. *ACM Queue* 06(04); <http://queue.acm.org/detail.cfm?id=1413263>
10. The OpenSSD project. [http://www.openssd-project.org/wiki/The\\_OpenSSD\\_Project](http://www.openssd-project.org/wiki/The_OpenSSD_Project)
11. PureStorage FlashArray. <http://www.purestorage.com/flash-array/purity.html>