An Investigation of Geographic Mapping Techniques for Internet Hosts

Venkata N. Padmanabhan[*]
Microsoft Research

Lakshminarayanan Subramanian[†]
University of California at Berkeley

ABSTRACT

In this paper, we ask whether it is possible to build an IP address to geographic location mapping service for Internet hosts. Such a service would enable a large and interesting class of location-aware applications. This is a challenging problem because an IP address does not inherently contain an indication of location.

We present and evaluate three distinct techniques, collectively referred to as *IP2Geo*, for determining the geographic location of Internet hosts. The first technique, *GeoTrack*, infers location based on the DNS names of the target host or other nearby network nodes. The second technique, *GeoPing*, uses network delay measurements from geographically distributed locations to deduce the coordinates of the target host. The third technique, *GeoCluster*, combines partial (and possibly inaccurate) host-to-location mapping information and BGP prefix information to infer the location of the target host. Using extensive and varied data sets, we evaluate the performance of these techniques and identify fundamental challenges in deducing geographic location from the IP address of an Internet host.

1. INTRODUCTION

In this paper, we ask the question: is it possible to build an IP address to geographic location mapping service for Internet hosts? Given an IP address, the mapping service would return the geographic location of the host to which the IP address has been assigned. This is a challenging problem because an IP address does not inherently contain an indication of geographic location.

Building an IP address to location mapping service (the *location mapping* problem for short) is an interesting problem in its own right. Such a service would also enable a large and interesting class of location-aware applications for Internet hosts, just as systems such as GPS [6] have for mobile devices. By knowing the location of a client host, an application, such as a Web service, could send the user location-based targeted information on local events, regional weather, etc. (*targeted advertising*), classify users based on location (e.g., count "hits" based on the region the user is located in), or control the availability of data based on user location (*territorial rights management* akin to TV broadcast rights). Each application may have a different requirement on the resolution of location information needed.

In this paper, we present several novel techniques, collectively referred to as *IP2Geo*, that approach the location mapping problem from different angles. These techniques exploit various properties of and observations on the Internet such as hierarchical addressing and correlation between delay and distance. We have analyzed a variety of data sets both to refine these techniques and evaluate their performance. To the best of our knowledge, ours is the first research effort in the open literature that studies this problem in detail.

The first technique, *GeoTrack*, tries to infer location based on the DNS names of the target host or other nearby network nodes. The DNS name of an Internet host sometimes contains clues about the host's location. Such a clue, when present, could indicate location at different levels of granularity such as city (e.g., *corerouter1.SanFrancisco.cw.net* indicates the city of San Francisco), state (e.g., *www.state.ca.us* indicates the state of California), or country (e.g., *www.un.cm* indicates the country of Cameroon).

The second technique, *GeoPing*, uses network delay measurements made from geographically distributed locations to infer the coordinates of the target host. It is based on the premise that the delay experienced by packets traveling between a pair of hosts in the network is, to first order, a function of the geographic separation between the hosts (akin to the relationship between signal strength and distance exploited by wireless user positioning systems such as RADAR[1]). This is, of course, only an approximation. So our delay-based technique relies heavily on empirical measurements of network delay, as discussed in Section 5.

The third technique, *GeoCluster*, combines partial (and possibly inaccurate) IP-to-location mapping information with BGP prefix information to infer the location of the host of interest. For our research, we obtained the host-to-location mapping information from a variety of sources, including a popular Web-based email site, a business Web hosting site, and an online TV guide site. The data thus obtained is *partial* in the sense that it only includes a relatively small

[*] http://www.research.microsoft.com/~padmanab/

[†] http://www.cs.berkeley.edu/~lakme/. The author was an intern at Microsoft Research through much of this work.

SIGCOMM'01, August 27-31, 2001, San Diego, California, USA..
Copyright 2001 ACM 1-58113-411-8/01/0008 ...$5.00.

number of IP addresses. We use BGP prefix information to expand the coverage of this data by identifying clusters of IP addresses that are likely to be located in the same geographic area. This technique is self-calibrating in that it can offer an indication of how accurate a specific location estimate is likely to be.

We have evaluated these techniques using extensive and varied data sets. While none of the techniques is perfect, their performance is encouraging. The median error in our location estimate varies from 28 km to several hundred kilometers depending on the technique used and the nature of the hosts being located (e.g., well-connected clients versus proxy clients). We believe that a significant contribution of our work is a systematic study of a broad spectrum of techniques and a discussion of the fundamental challenges in determining location based just on the IP address of a host.

The rest of this paper is organized as follows. In Section 2 we survey related work. In Section 3 we describe our design rationale and experimental methodology. We present the details of the three IP2Geo techniques and an analysis of their performance in Sections 4, 5, and 6. Finally, we present a summary and discuss the contributions of our work in Section 7.

2. RELATED WORK

There has been been much work on the problem of locating hosts in wireless environments. The most well-known among these is the Global Positioning System (GPS) [6]. However, GPS is ineffective indoors. There have been several systems targeted specifically at indoor environments, including Active Badge [9], Bat [10], and RADAR [1]. As we discuss later, our GeoPing technique uses a variant of one of the algorithms we had developed for RADAR. However, in general these techniques are specific to wireless networks and do not readily extend to the Internet.

In the Internet context, an approach that has been used to determine location is to seek the user's input (e.g., by requiring the user to register with and/or log in to the site, by storing the user's credentials in client-based cookies, etc.). However, such approaches are likely to be (a) burdensome on the user, (b) ineffective if the user uses a client other that the one where the cookie is stored, and (c) prone to errors due to (possibly deliberate) inaccuracies in the location information provided by an *individual* user. (In Section 6, we discuss how GeoCluster deals with such inaccuracies by aggregating information derived from individual users.)

An alternative approach is to build a service that maps an IP address to the corresponding geographic location [16]. There are several ways of doing this:

1. Incorporating location information (e.g., latitude and longitude) in Domain Name System (DNS) records.

2. Using the *Whois* [8] database to determine the location of the organization to which an IP address was assigned.

3. Using the *traceroute* [11] tool and mapping the router names in the path to geographic locations.

4. Doing an exhaustive tabulation IP address ranges and their corresponding locations.

The DNS-based approach was proposed in RFC 1876 [17]. This work defines the format of a new Resource Record (RR) for the DNS, and reserves a corresponding DNS type mnemonic (LOC) and numerical code (29). The DNS-based approach faces deployment hurdles since it requires a modification of the record structure of the DNS records. This also burdens administrators with the task of entering the LOC records. Moreover, there is no easy way of verifying the accuracy of the location entered.

An approach used widely in many tools is to query Whois servers [8]. Tools such as IP2LL [26] and NetGeo [14] use the location information recorded in the Whois database to infer the geographic location of a host.

There are several problems with Whois-based approaches. First, the information recorded in the Whois database may be inaccurate or stale. Also, there may be inconsistencies between multiple servers that contain records corresponding to an IP address block. Second, a large (and geographically dispersed) block of IP addresses may be allocated to a single entity and the Whois database may contain just a single entry for the entire block. For example, the 4.0.0.0/8 IP address block is allocated to BBN Planet (now known as Genuity) and a query to ARIN Whois database returns the location as Cambridge, MA for any IP address within this range.

An alternative approach is based on the traceroute tool. The basic idea here is to perform a traceroute from a source to the target IP address and infer location information from the DNS names of routers along the path. A router name may not always contain location information. Even when it does, it is often challenging to identify the location information since there is no standard naming convention that is used by all ISPs. We discuss these issues in more detail when we present GeoTrack in Section 4. Examples of location mapping tools based on traceroute include VisualRoute [31], Neotrace [29], and GTrace [15].

Finally, there are location mapping services, such as EdgeScape from Akamai [18] and TraceWare from Digital Island [22]. Given the extensive relationship that these large content distribution networks enjoy with several ISPs, it is conceivable that these location mapping services are based on an exhaustive tabulation of IP address ranges and the corresponding location. However, the algorithms employed by EdgeScape and TraceWare are proprietary, so it is difficult for us to compare them to our research effort.

2.1 Fundamental Limitation due to Proxies

Many Web clients are behind proxies or firewalls. So the "client" IP address seen by the external network may actually correspond to a proxy, which may be problematic for location mapping. In some cases the client and the proxy may be in close proximity (e.g., a caching proxy on a university campus). However, in other cases they may be far apart. An example of the latter is the AOL network [19], which has a centralized cluster of proxies at one location (Virginia) for serving client hosts located all across the U.S. Figure 1 shows the cumulative distribution function (CDF) of the distance between the AOL proxies and clients. (The *likely* location of clients was inferred from the data sets described in Section 3.5.) We observe that a significant fraction of the clients are located several hundred to a few thousand kilometers from the proxies.

Proxies impose a fundamental limitation on all location

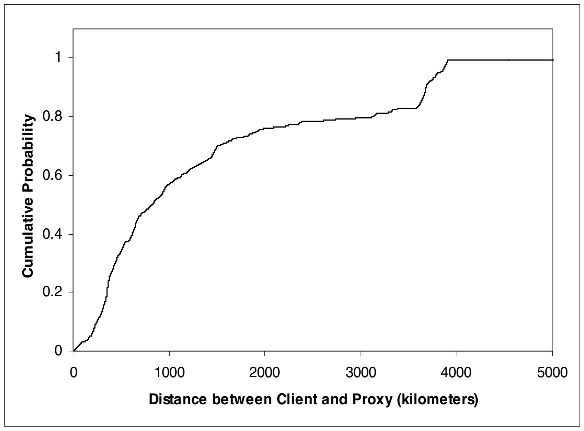

Figure 1: Distribution of distance between AOL proxies and clients.

mapping techniques that depend on client IP address. This includes techniques based on Whois, traceroute (e.g., GeoTrack), and network delay measurements (e.g., GeoPing). Not only are these schemes unable to determine the true location of a client, they are also oblivious to the error (i.e., these schemes would incorrectly return the location of the proxy without realizing the error). Our GeoCluster technique is an exception in that it is often able to automatically tell when its location estimate is likely to be erroneous. So rather than incorrectly deducing the location of the client based on the IP address of the proxy, GeoCluster would refrain from making a location estimate at all. We discuss this issue is more detail in Section 6.3.

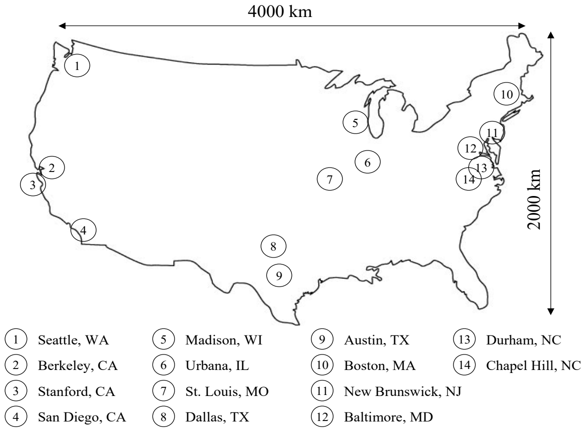

Figure 2: Outline map of the U.S. showing locations of our probe machines.

3. DESIGN RATIONALE AND EXPERIMENTAL METHODOLOGY

In this section, we first discuss the design rationale for IP2Geo in view of the limitations of existing techniques discussed above. We then describe the experimental setup and data sets used in our study.

3.1 Design Rationale

In Section 2, we discussed several existing approaches to location mapping and their limitations. Among these, Whois and traceroute based approaches are the most popular. To get a better understanding of the strengths and limitations of these two approaches, we have developed GeoTrack, a traceroute-based tool for performing location mapping that is largely based on the same principles as existing traceroute-based tools such as VisualRoute and GTrace. We compare the performance of GeoTrack with NetGeo, a Whois-based location mapping tool.

IP2Geo also includes two new techniques, GeoPing and GeoCluster, which operate very differently from existing approaches. GeoPing exploits the correlation between network delay and geographic distance to determine the location of a host. Although this correlation is not strong enough to be captured in a mathematical model, we show that is indeed possible to be build a coarse-grained location tracker using just delay measurements.

We also describe GeoCluster, a powerful new technique that combines partial IP-to-location mapping information obtained from a variety of sources and topological clustering data [12] to do location mapping. Our results indicate that GeoCluster performs the best among the IP2Geo techniques.

Before getting to the details of these techniques, we describe the experimental setup and data sets that we have used in our study.

3.2 Geographic Setting

All of our experiments are set in the United States (U.S.). The main reason for this restriction is that, as of the time of this writing, the bulk of the data sets and probe machines that we have pertain to or are located in the U.S. While there may be limitations to studying a single country, the U.S. still offers a large and varied testbed for our research. The U.S. consists of 50 states, 48 of which are located in the large geographic area depicted in Figure 2, and two others that are located 2000 km to the northwest and 4000 km to the southwest, respectively, of this landmass. (In addition, our data sets recorded the U.S. capital, Washington DC, as a separate entity, so we effectively had 51 "states".) Thus, the U.S. is as large as certain continents in terms of geographic expanse. It is also home to a sizeable fraction of the Internet, in terms of networks, routers, end hosts, and users. So we believe the research reported in this paper is interesting despite being limited to the U.S.

3.3 Probe Machines

We obtained access to probe machines at the 14 locations depicted in Figure 2. These machines were distributed geographically across the U.S. All of them were well-connected hosts on university campuses except for the machine at Seattle, WA, which was located at a corporate site (Microsoft). These probe machines were used to make delay measurements for GeoPing and to initiate traceroutes for GeoTrack.

As we explain later in Section *5.1.1*, GeoPing is primed using a database of delay measurements from the probe machines to several "target" machines at known locations. To obtain such a database, we constructed a list of 265 Web servers (termed *UnivHosts*) spread across university campuses in 44 states of the U.S. The selection of university servers as target hosts offered the advantage that we were quite certain of their actual geographic location.

The UnivHosts data set is also used to evaluate the performance of GeoTrack and GeoCluster.

3.4 BGP Data

BGP routing information was derived from dumps taken at two routers at BBN Planet [20] and MERIT [28]. Since GeoCluster only requires the *address prefix (AP)* information, we constructed a superset containing address prefix information derived from both sources. In all there were 100,666 APs in our list.

3.5 Partial Location Mapping Information

We obtained partial IP-to-location mapping information from three sources. The data sets we obtained were partial in the sense that they only covered a small fraction of IP address space in use. Note that in no case did we have access to user IDs or other user-specific information. Our data sets only contained IP address and location information. So our work did not compromise user privacy in any way.

1. *Hotmail*: Hotmail [24] is a popular Web-based email service with several million active users. Of the over 1 million (anonymous) users we obtained information for, we focused on the 417721 users who had registered their location as being in the U.S. The location information we obtained from the users' registration records was at the granularity of U.S. states. In addition, we obtained a log of the client IP addresses corresponding to the 10 most recent user logins (primarily in the first half of 2000). We combined the login and registration information to obtain a partial IP-to-location mapping.

2. *bCentral*: bCentral [21] is a business Web hosting site. Location information at the granularity of zip codes was derived from HTTP cookies. In all we obtained location information corresponding to 181246 unique IP addresses seen during (part of) a day in October 2000.

3. *FooTV*: FooTV is an online TV program guide where people look up program listings for specific zip codes. (We do not reveal the name of the site here due to anonymity requirements.) From traces gathered over a two-day period in February 2000, we obtained a list of 142807 unique client IP addresses and 336181 (IP,zip) pairs corresponding to the client IP address and the zip code that the user specified in his/her query. A subset of the IP addresses had more than one corresponding zip code, which were usually clustered together geographically.

In the case of bCentral and FooTV, we mapped the zip code information to the corresponding (approximate) latitude and longitude using information from the U.S. Census Bureau [30]. In the case of Hotmail, we computed the *zip-center* of each state by averaging the coordinates of the zip codes contained within that state.

The partial IP-to-location mapping obtained from these sources may contain inaccuracies. For instance, in the case of Hotmail and bCentral users may have registered incorrect location information or may connect from locations other than the one they registered. In the case of FooTV, users may enquire about TV programs in areas far removed from their current location, although we believe this is unlikely. Regardless, we explain in Section 6 how GeoCluster is robust to such inaccuracies in location information.

4. THE GEOTRACK TECHNIQUE

The GeoTrack technique tries to infer location based on the DNS names of the host of interest or other nearby network nodes. Network operators often assign geographically meaningful names to routers[1], presumably for administrative convenience. For example, the name *corerouter1.SanFrancisco.cw.net* corresponds to a router located in San Francisco. We stress that having geographically meaningful router names is *not* a requirement or a fundamental property of the Internet. Rather it simply an observation that is generally supported by empirical data.

We define a router to be *recognizable* if its geographic location can be inferred from its DNS name. Routers whose IP address cannot be mapped to a DNS name or whose DNS name does not contain meaningful location information are considered as not being recognizable.

GeoTrack uses these geographic hints to estimate the location of the target host. First, it determines the network path between a probe machine and the target host using the traceroute tool. Traceroute reports the DNS names of the intermediate routers where possible. Then GeoTrack extracts location information from the DNS names of recognizable routers along the path. Thus, it traces the *geographic path* to the target host. Finally, GeoTrack estimates the location of the target host as that of the last recognizable router in the path (i.e., the one closest to the target).

As noted in Section 2, traceroute-based approaches that extract geographic hints from router names have been proposed before (e.g., GTrace [15], VisualRoute [31]). However, we are not aware of work in the open literature on a quantitative evaluation of the traceroute-based approach to determining the geographic location of hosts. Our goal is precisely to do such an evaluation. Due to the logistic difficulties associated with obtaining and running existing traceroute-based tools, we decided to write our own tool based on GeoTrack to do large-scale experimentation. We have tested our tool over a large sample of IP addresses and found that its coverage is comparable to VisualRoute within the U.S. and in Europe.

4.1 Extracting Geographic Information from Router Names

Geographic information is typically embedded in the DNS name of a router in the form of a *code*, which is usually an abbreviation for a city, state, or country name. There is no standard naming convention for these codes. Each ISP tends to use its own naming convention. This makes the task of extracting location information from DNS names challenging.

Based on empirical data, we have observed that there are basically three types of codes that indicate location: city codes, airport codes, and country codes. Some ISPs assign DNS names to routers based on the airport code of the city they are located in. Since airport codes are a worldwide standard, such a naming convention greatly eases the task

[1] To be precise, DNS names are associated with router *interfaces*, not routers themselves. However, for ease of exposition we only use the term "router".

of determining the router's location. For example *sjc2-cw-oc3.sjc.above.net* refers to a router in San Jose, CA (airport code *sjc*). However, many ISPs use non-standard codes for cities. We have noticed that the city of Chicago, IL has at least 12 different codes associated with it (e.g., *chcg, chcgil, cgcil, chi, chicago*). We have also observed that many routers outside the United States have the country codes embedded in their names. For example, the router with the name *asd-nr16.nl.kpnqwest.net* is located in the Netherlands (country code *nl*). The country information can be very useful in (partially) validating the correctness of the location guessed based on city or airport codes.

We examined several thousand distinct router names encountered in the large set of traceroutes that we performed from our 14 probe locations. We compiled a list of approximately 2000 airport and city codes for cities in the U.S. and in Europe. Of the entire set of airport codes [27], our list only includes a relatively small fraction of codes that are actually used in router names. Since GeoTrack deduces location by doing a string match of router names against the codes, constructing a list with as few superfluous codes as possible decreases the chances of an inadvertent match.

To further reduce the chances of an inadvertent match, we divided the list of location codes into separate pieces corresponding to each major ISP (e.g., AT&T, Sprint, etc.). When trying to infer location from a router name associated with a particular ISP, GeoTrack only considers the codes in the corresponding subset.

There is the question of how router names are matched against the location codes. Simply trying to do a string match without regard to position of the matching substring may be inappropriate. For example, the code *charlotte*, which corresponds to Charlotte, NC in the eastern U.S., would incorrectly match against the name *charlotte.ucsd.edu*, which corresponds to a host in San Diego, CA in the western U.S. Through empirical observation, we have defined ISP-specific parsing rules that specify the position at which the location code, if any, must appear in router names associated with a particular ISP. We split the router name into multiple pieces separated by dots. The ISP-specific parsing rules specify which piece(s) should be considered when looking for a match. For example, the rule for Sprintlink specifies that the location code, if present, will only be in the first piece from the left (e.g., *sl-bb10-sea-9-0.sprintlink.net* containing the code *sea* for Seattle). The rule for AlterNet (UUNET) specifies that the code, if present, will only appear in the third piece from the right (e.g., *192.atm4-0.sr1.atl5.alter.net* containing the code *atl* for Atlanta).

4.2 Performance Evaluation

We compare the performance of GeoTrack and a Whois-based tool, NetGeo [14], both for university hosts drawn from the UnivHosts data set and for a more diverse set of hosts drawn from the FooTV data set. The latter consists of a random sample of 2380 client IP addresses drawn from the FooTV data set. While many of the FooTV clients connected via proxies, none of the university hosts was behind a proxy. For this experiment, we used the probe machine at UNC in Raleigh, NC as the source of all traceroutes.

We quantify the accuracy of a location estimate using the *error distance*, which we define as the geographic distance between the actual location of the destination host and the estimated location. In the case of FooTV, the "actual" location corresponds to the zip code recorded in the FooTV data set which, as noted in Section 3.5, may not be entirely accurate. Also, an IP address may be associated with multiple locations, either because it was allocated dynamically (say using DHCP [5]) or because it belonged to a proxy host (such as a Web proxy or a firewall). GeoTrack, on the other hand, would only make a single location estimate for a particular IP address. In our evaluation, we compute separate error distances corresponding to the many "actual" locations associated with an IP address.

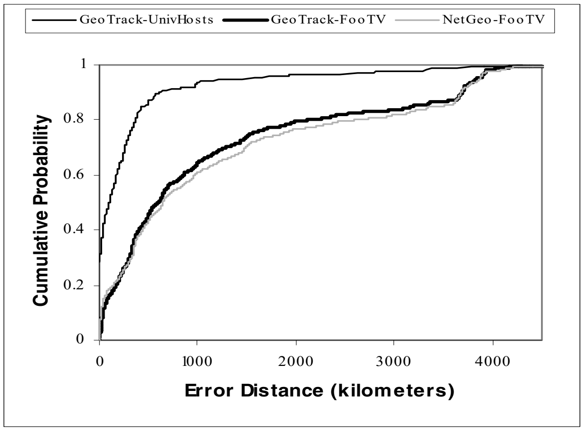

Figure 3: CDF of the error distance for GeoTrack and NetGeo.

Figure 3 shows the CDF of error distance for both GeoTrack and NetGeo. It is very interesting to note the similarity between the "NetGeo-FooTV" and "GeoTrack-FooTV" curves beyond the 70th percentile mark, and the distribution of distance of AOL clients from their proxies in Figure 1. GeoTrack determines the location of the AOL proxies as Washington, DC while NetGeo returns the location as Sterling, VA. The similarity in the curves can be attributed to the fact that these two locations are only about 35 km apart. (Moreover, AOL's proxies are also located in the same vicinity.)

We also observe that the performance of GeoTrack is only slightly better than that of NetGeo. GeoTrack exhibits a median error distance of 590 km and NetGeo a median of 650 km. Since many of the FooTV clients are behind proxies, neither Geotrack nor NetGeo is able to estimate the client's location accurately.

It is interesting to note that there is a significant difference in the performance of GeoTrack for the well-connected UnivHosts hosts as compared to that for FooTV clients. For instance, the median error distance is 102 km for the former while is is 590 km for the latter. The reason for this difference is that (a) none of the hosts in UnivHosts is behind a proxy, and (b) these hosts are well connected in the sense that a traceroute to them generally completes and yields a last recognizable router that tends to be close to the target host.

5. THE GEOPING TECHNIQUE

The GeoPing technique seeks to determine the geographic location of an Internet host by exploiting the relationship between network delay and geographic distance. GeoPing measures the delay to the target host from multiple sources

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**