

Public Health for the Internet (φ)

Towards A New Grand Challenge for Information Management

Joseph M. Hellerstein* Tyson Condie* Minos Garofalakis^{†*} Boon Thau Loo[†]
Petros Maniatis[†] Timothy Roscoe^{†‡} Nina A. Taft[†]

*UC Berkeley, [†]Intel Research Berkeley, [‡]ETH Zürich, [†]U. Pennsylvania

ABSTRACT

Business incentives have brought us within a small factor of achieving the database community’s Grand Challenge set out in the Asilomar Report of 1998. This paper makes the case for a new, focused Grand Challenge: Public Health for the Internet. The goal of PHI (or φ) is to enable collectives of hosts on the Internet to jointly monitor and promote network health by sharing information on network conditions in a peer-to-peer fashion. We argue that this will be a positive effort for the research community for a variety of reasons, both in terms of its technical reach and its societal impact.

This version of the φ vision is targeted at readers in the database research community, but the effort is clearly multidisciplinary. A more generalist version of this paper will be maintained at <http://openphi.net>.

1. RALLYING A DIVERSE COMMUNITY

We recommend a ten-year goal for the database research community: . . . Make it easy for everyone to store, organize, access, and analyze the majority of human information online.

– The Asilomar Report on Database Research

Google’s mission is to organize the world’s information and make it universally accessible and useful.

– Google Corporation

The database research community is well-regarded in computer science circles for its focus and relevance. Part of this reputation comes from the periodic reports reflecting the consensus of community leaders, which serve both to rally the community internally, and to present a united front to colleagues, funding agencies, and industrial partners. Today, the community’s focus and relevance are being challenged on two fronts: the technology business climate, and the academic research climate.

In 1998, the *Asilomar Report* laid out a Grand Challenge

This article is published under a Creative Commons License Agreement (<http://creativecommons.org/licenses/by/2.5/>.) You may copy, distribute, display, and perform the work, make derivative works and make commercial use of the work, but, you must attribute the work to the author and CIDR 2007.

^{3rd} Biennial Conference on Innovative Data Systems Research (CIDR) January 7-10, 2007, Asilomar, California, USA.

for the community, quoted above. Nine years later, this challenge is remarkably close to the corporate missions of major Internet services, which are widely viewed – even within much of the computer science community – as having made significant progress toward addressing the challenge. Some database cognoscenti quibble with the completeness of these efforts to date, particularly as regards structured data. However, a number of other well-known database researchers are voting with their feet, leaving prestigious academic and industrial research posts to join the staff at these companies. Efforts toward publishing, hosting and integrating structured data on the web are well underway. In short, even if one does not believe that the Asilomar Challenge is completely solved, it appears that the goal of enormous advertising revenue will drive this agenda forward naturally and aggressively. This is no longer a Grand Challenge for Science. This is Business, and it is in full swing.

In parallel with this development, the last decade has seen remarkable growth in the database research community, both demographically and intellectually. Universities in Canada, Singapore, and India now boast database groups bigger than the traditional leaders in the U.S. and Europe, and produce quality research publications and students accordingly. Meanwhile, the database community’s technical focus has grown to overlap significantly with areas as diverse as statistical machine learning, networking, security, and computational biology, in addition to traditional overlaps with operating systems, programming languages, computer architecture, and theoretical computer science. These developments are both exciting and welcome. But they significantly strain the community’s traditional ability to achieve research focus and, by doing so, to forcefully drive technology development and transfer.

The Asilomar Report was the last major effort by the community to produce a Grand Challenge, however vague¹. In this paper we begin a discussion of a new agenda we call Public Health for the Internet (PHI or φ). We hope that φ can serve as one of a few rallying points for the growing community of database researchers, based on important technical and societal agendas.

¹The *Lowell Report* in 2003 [1] included many technical challenges, but no single motivating Grand Challenge.

1.1 Parameters of the Grand Challenge

Every generation needs a new revolution.
– Thomas Jefferson.

A successful Grand Challenge must inspire and focus the community of scientists and engineers. Before proceeding to describe our own proposal, we sketch some considerations that we think are useful for today’s community of Information Management researchers. These stand in contrast to earlier challenges in database research, both in motivation and in the technical agenda.

1.1.1 Motivation: The Common Good

Departing from the roots of database research in business data processing, we would like to see the next Grand Challenges have a direct, positive outcome for society. The growing scientific community in this area could use a motivation beyond advertising and service revenues, which have lately become the main drivers for the software industry. Arguably the best such motivation is one that affects the daily lives of ordinary people, starting with the scientists themselves, expanding out through their peers and families to the community at large. It is also important that the challenge have a crisp mission statement, which can engage both technical and non-technical audiences – from funding agencies to taxpayers to school children. Of course, commercial benefits are likely to ensue if the social benefits are compelling. After a period of open innovation, one efficient means for promoting the common good may be to harness the efficiency of the marketplace.

1.1.2 Technical Parameters

All Grand Challenges need to seek fertile ground between the clearly trivial and the truly impossible. To draw in the breadth of today’s diverse database research community, any new Grand Challenge should have many points of attack from a wide technical territory, differing in the dimensions of both technical areas and research methodologies. To focus this discussion, we highlight three distinguishing “revolutionary” themes for the next Grand Challenges:

Architecture: *The Revolution Will Not Be Centralized*²

As noted above, there are already strong incentives for an individual business (and, more quietly, an individual government) to collect, crawl, index and mine large volumes of information. To exercise new architectural and algorithmic ideas, a revolutionary system should not run in a single enterprise’s hosting facility. It should instead target massive distribution and the participation of multiple parties. These requirements are not exotic. They arise in a variety of settings that involve some combination of (1) distributed data gathering, (2) real-time querying, triggering and dissemination, and/or (3) limited trust and access due to diverse economic or political goals.

Incentives: *The People, United, Will Never Be Defeated*³

To drive a massively decentralized revolution, three factors must come into play. First, the application has to be attractive to a large, geographically distributed user base who

²“The Revolution Will Not Be Televised” – Gil Scott-Heron, 1971.

³“El pueblo unido jamás será vencido” – Sergio Ortega, 1973.

will want to provide data and a workload. Second, the information in aggregate has to be more powerful than it is when isolated. Finally, that power should accrue directly to the parties providing the data. Otherwise other economic means (payment, free services) have to be used to provide incentives, and these approaches are typically too indirect and complex to be used in framing a Grand Challenge for Science.

Information: *Information is not Knowledge*⁴. *Knowledge is Power*⁵. *With great power comes great responsibility*⁶.

Efforts to unify information from many parties are inherently messy. On one hand, there is the difficulty of the problem: information at a global scale is always incomplete, rife with inaccuracies and misrepresentations, and varied in its representation. On the other hand, there is the dangerous potential power of a solution: the goal of personal privacy seems to work against a vision of exposing massive amounts of information to the public. In our view, a new Grand Challenge for Information Management should encompass these important problems, and encourage researchers to seek firm and fertile ground between the quicksand of the technical and societal challenges involved in massive information sharing. The Grand Challenge should entail a sufficiently structured environment so that integrating information and inferring knowledge are challenging but tractable. It should also include the problem of protecting and validating the preservation of individual privacy.

Given a Grand Challenge with these parameters, many of the community’s current research topics could be brought to bear: data mining, data privacy, distributed query processing and optimization, stream and event processing, data integration, data reduction, probabilistic data and inference, and so on. Moreover, there is room for contributions from multiple research styles, including system building, performance analysis, and theoretical investigation. What remains is the need for a specific, focused Grand-Challenge Statement that will rally the community to collectively target its skills to specific goals, and have a yardstick by which to measure progress.

2. PUBLIC HEALTH FOR THE INTERNET

With this introduction, we now turn our attention to our specific proposal for a new Grand Challenge. We begin with a simple mission statement, intended to convey the main idea of the challenge to a casual listener. We then move on to a somewhat more nuanced and detailed description.

2.1 The Elevator Pitch

The φ Grand Challenge: *Computers on the Internet should organize themselves into a worldwide community watch, tracking and containing the spread of viruses, worms, spyware and other harmful traffic. This should be done without sacrificing end-user privacy or autonomy, and without placing undue responsibility or control into the hands of any one party.*

⁴Attributed to Albert Einstein. Also, “Information is not knowledge. Knowledge is not wisdom. Wisdom is not truth. Truth is not beauty. Beauty is not love. Love is not music. Music is the best.” – Frank Zappa, *Joe’s Garage*, 1979.

⁵Sir Francis Bacon, *Meditationes Sacrae*, 1597

⁶Uncle Ben, *Spider-Man*, 2002.

2.2 The One-Pager

Internet security is perhaps the key challenge facing computing today. It has been estimated that Internet “malware” (viruses, worms, spyware and the like) cost global businesses between 169 and 204 billion dollars in 2004⁷, despite significant rollouts of Internet security software [14]. It has also been convincingly argued that the entire Internet could be taken over by a worm in just minutes [28].

Internet security is often cast in terms of medical analogies: viruses, vaccines and the like. Medicine is a useful analogy, but as a discipline, medicine works by curing *individuals*, and focuses less on widespread problems. The complementary field of Public Health takes a more population-focused, community-oriented approach. Public Health research often focuses on understanding and mitigating the spread of disease in a population. The Internet is a shared medium, and it is time that Internet security shifts to analogies from Public Health rather than Medicine.

Public Health for the Internet (PHI, or φ) is a scientific challenge for the computer science community. The goal of PHI is to enable machines on the Internet to band together into a “community health watch” infrastructure that aggressively shares, analyzes, and acts upon information observed at the various machines. The end result should be a global Internet that is smarter and stronger than any subset of compromised or malicious nodes within it, while ensuring the efficiency, privacy and autonomy that users expect of a global, public medium.

The technical key to the PHI vision is the sharing of rich, rapidly-evolving information across the masses of computers connected to the Internet. It is becoming increasingly apparent that the only reliable place to check for malicious traffic is at the endpoint destinations of the traffic. Traditionally, networks depended on “border security” such as firewalls, which funneled traffic through checkpoints that could prevent unwanted traffic from entering sheltered subnets. This approach has broken down with the prevalence of mobile computers that bring in malware from outside the firewall, and web services like email that “tunnel” malicious traffic through firewalls.

PHI would allow endpoint traffic observations to be shared across machines for the common good, enabling all users to benefit from the collective information gathered by the full population. To be effective, this information would have to be shared extremely quickly: fast enough to keep up with malicious network traffic. It would also have to be shared securely, so that attackers would be unable to bypass the defense mechanisms, or subvert their purpose and turn them into counter-intelligence services or even weapons of attack. Sophisticated analyses would have to be run on the data and the results shared in real time. Finally, all sharing of information would need to be done without compromising the privacy and autonomy of the users of the network.

One proposal for achieving this vision is to build a “Center for Disease Control” for the Internet [28] that gathers reports from all the machines on the network, analyzes them,

⁷N.B.: Roughly an order of magnitude more than database industry revenues.

and disseminates warnings or commands. The problem with this approach is basic: who runs the Center, and what do they Control? In an era of increasing concern over government subpoenas, individual privacy, and corporate liability, the idea of centralizing Internet security in the hands of a few large organizations seems politically intractable.

Instead, we believe the time is ripe, both technically and politically, to get the Internet’s end users to band together in a global *peer-to-peer* (P2P) system that monitors the Internet at scale – with neither a Center nor any explicit Control of the system. The incentives for end users are clear: more security and hence less hassle, without having to trust a central authority. There are also incentives for vendors of the end users’ PCs, operating systems and home routers, who are looking to differentiate their products with new network security features. Finally, enterprises can use PHI technology internally as a way to replace their crumbling firewalls.

As an added incentive for end users, PHI software could provide them with better understanding of their local experience by reference to the global, shared information: for example, to compare whether the sites attacking them are also attacking others, and characterize what aspects of their system are attracting attack. This can allow users to better answer questions like “*Why is my computer behaving strangely?*”, “*Am I being targeted?*”, “*Is this machine contacting me a ‘bad guy’?*”, or even “*Is it really critical that I upgrade my software?*”

In our vision, the φ software base would consist of (a) a variety of network “sensor” software modules at the endpoints that gather and ingest key network security and performance characteristics at each endpoint; (b) a peer-to-peer protocol and compliant software that would allow machines to collaboratively analyze this information across the collective, disseminating important information, triggering automated responses, and answering user queries; and (c) useful and attractive end-user interfaces – both to incent users to join the collective, and to encourage them to be more proactive about securing their own machines.

3. A SAMPLING OF RESEARCH ISSUES

The φ Grand Challenge is hard, and parts of it will be unfamiliar to a database audience. For example, some low-level aspects of it have to do with the internals of Internet protocols that do not significantly touch on the core competencies of this community. However, a key component of the φ challenge involves distributed, real-time information management. The traditional database research community could profitably set many of its current areas of research in this context, and in doing so make a tangible difference in the computing experience of users worldwide. In this section, we highlight a number of research problems that have occurred to us in our initial efforts. We also fully expect that there are many other issues we have not even begun to consider. Indeed, our hope is that this is the beginning of a conversation with many participants.

3.1 Massively Distributed Stream Processing

The driving data model for a φ system will inevitably be one of distributed streams of data coupled with historical repositories. The streams will come from a variety of distributed

sources, including packet traces, firewall log events, operating system logs, and possibly the logs of popular networked applications like browsers, communication tools (chat, network telephony), online media players, and peer-to-peer applications. There is also a role for stored data, which may be at the endpoints, but which also may come from network infrastructure, including routers, Internet “Weather Services,” and potentially shared archives of prior observations.

The collective system needs to be able to process such streams of data in a variety of ways. For the most part, we do not expect users to explicitly interrogate the health of the network on a regular basis. Instead, the information processing in a φ system will mostly be data- and event-driven. This requires communication-efficient techniques for massively distributed, shared “triggers” or “event processing.”

As a modest example, we consider the problem of quickly determining the origin of a network worm. An incoming network packet at one node might trigger a traditional local firewall rule that recognizes the worm. Then the system might autonomously engage the packet’s sender in a backtrace of the origin of the worm. This has been shown effective using centralized packet traces [30]. In the φ context, this amounts to an Event-Condition-Action rule in which the Event detection (packet arrival) and the Condition checking (firewall rules) are local network logic, and the Action is a distributed transitive closure query that computes the ancestor relation for the packet across multiple nodes.

A more complicated family of triggers requires that the condition in the Event-Condition-Action cycle be checked *globally* in the face of frequent events. As an example, consider a distributed traffic governor that tries to ensure that a large coalition of compromised PCs (a so-called “botnet”) cannot be utilized for Distributed Denial of Service (DDoS) attacks – such logic could run reliably on network interface cards, for example. This governor would need to trigger if an aggregate of traffic across all nodes goes above a threshold for any destination IP address on the Internet; in a SQL-like syntax, this condition would be

```
SELECT destination FROM packetstream
GROUP BY destination
HAVING COUNT(packets) > X
```

Efficiently tracking such aggregate quantities across distributed nodes is a difficult challenge. It has been addressed algorithmically by a number of recent database and networking papers [2,3,13], but to date these ideas have not been seriously explored in a workable system.

Another recent example arises in the analysis of the network-level behavior of spammers. It is commonly thought that modern spam is largely delivered by botnets. In a forensic study of archived data, Ramachandran and Feamster validated this belief, and found that the distribution of infected bot machines in the network was non-uniform and detectable, even though activity as a spammer is transient. They hypothesize that “the uneven distribution of spam and botnet activity across IP space – and the differences in this distribution from legitimate email – suggests that spam filters and intrusion detection systems might monitor network-wide spam arrival patterns for changes in these distributions

to detect anomalies such as a surge in spam activity.” [23] Achieving this goal requires near-real-time distributed monitoring and querying to compute a distribution of hosts sending spam; this distribution is laid out on the hierarchical ID space of Internet addresses, which has particular properties that motivate specialized data reduction techniques [25].

More recently, it has been suggested that distributed tracking of simple aggregates like COUNTs can be used as a kernel for approximately tracking more sophisticated statistical shifts in network traffic [10]. The challenge here is more like real-time distributed data mining: to identify “surprising” shifts in the patterns of Internet traffic, which may indicate the emergence of new as-yet-unrecognized worms, the triggering of botnet attacks, or simply a shift in the Zeitgeist of Internet users. Recognizing and responding to these shifts is an important part of making a φ infrastructure resilient to the shifting nature of the network and its population.

These problems drive a new generation of database query processing challenges, which will require a combination of distributed system design, efficient approximate query processing, and a new class of multiquery optimization and information dissemination that can collectively monitor and report on variants of such queries for millions of end users. The surface of this problem has been scratched, and it is clear there is much more to be done.

3.2 Network Design for Distributed Queries

Any massively distributed system needs to coordinate the communications of its members. That means it must find efficient ways to dynamically track the addresses of the participants as they come and go (network “churn”), and route communications between pairs of participants that may not know of each other’s presence. This is the domain of *overlay networks*, which has received significant research and commercial attention in recent years.

While this topic has been explored largely outside the database research community, P2P filesharing explicitly blurred the distinction between network routing and distributed queries. P2P filesharing treats routing as an entirely content-based problem: query messages should be routed to nodes that have matching data. It similarly blurs network “topology” (the shape of the network graph) with querying: the overlay network graph should try and keep nodes with related content and queries well connected to each other. This idea was addressed more formally for doing content-based routing for equality predicates, in the *Distributed Hash Table* (DHT) literature [24,26,29], as well as in various proposals for range predicates. In essence, overlay network design has evolved into a distributed indexing problem, which has caught the attention of a breadth of researchers in databases, networks, systems and algorithms.

If there is one key lesson from the early days of query optimization, it is that no single index structure or join algorithm is always best, and that a good system should employ a family of indexes and algorithms, choosing among them to best suit a workload. This idea has yet to be explored seriously in the context of distributed queries. In our own work on the PIER engine, we took the route of trying to achieve multiple behaviors on a single DHT network [11].

An intriguing alternative is to return to database systems roots, and maintain – or perhaps construct on-the-fly – a set of many custom overlay networks to suit particular queries or workloads. For example consider a query with an equality join and an aggregation. Gossip could be used for query dissemination, a DHT could be used for the join, and a custom tree could be built for performing the aggregation. Ideally, of course, these decisions are made by a query optimizer, which for continuous queries would also need to adaptively change the plan in response to changing conditions. In essence this agenda couples database query optimization with overlay network design. The problem becomes only richer and more interesting when multiple ongoing queries are taken into account, which is the realistic setting for φ .

One direction we have pursued in this regard is *Declarative Networking*, in which a recursive query language is used to specify the implementation of an overlay network [17–19]. To date we have demonstrated that declarative languages for overlay networks can specify a broad range of overlay network protocols in *orders of magnitude less code* than traditional programming languages. As one example, we have a running implementation of the Chord overlay [29] that consists of a 41-line logic program, running within our *P2* system for declarative networking [18]. Our implementation in the *P2* system compiles that program down to a dataflow graph akin to a query plan, which results in an executable that combines the logic of a configurable network router like Click [16] with a networked query engine like PIER [11]. Not only does this bring the power of declarative languages to the construction of distributed systems, it opens up the possibility of automatically co-optimizing the overlay network and the query functionality in a single framework.

3.3 Privacy and Security

Privacy has been a hot topic in the database community in recent years. But it is a slippery topic to scope carefully. In some contexts, legal frameworks provide guidelines – the European Union has extensive rules on data privacy, and the US has policies in place for some domains like healthcare.

A key challenge in defining “useful” digital privacy work is to understand what people actually want. Indeed, most people can only express this by discussing examples of information they would not like to reveal to particular parties. Undoubtedly the best way to get ground truth while respecting people’s privacy desires is to offer them the opportunity to share their data on their own terms. This is not the kind of experiment that is easy for scientists to arrange at all, let alone in a way that sidesteps liability issues.

P2P systems offer such an opportunity. However, there are few *P2P* systems that promote a social good; SETI@Home may be the best example, but it is computation-intensive rather than data-intensive, and hence has minimal privacy implications. φ is a setting in which computer scientists interested in protecting privacy can engage with users who are interested – for the best of reasons – in sharing their private information, but in a variety of controlled ways. The application requires the development of policies and algorithms for guaranteeing limited exposure, and for allowing users to verify adherence to protocol. As a form of bootstrapping, very simple controls and sharing incentives can be offered to

open-source φ software users who “opt in” to the system. But if the technology is to eventually be bundled in products (e.g., network cards or routers), more robust techniques will be required to maximize user trust and minimize vendor liability.

3.3.1 Verifiability and other Security Issues

The database community’s efforts on privacy have largely addressed the security of stored data, and the way data is exposed – to queries, and to federated query engines during query processing.

Somewhat surprisingly, the community has all but ignored the question of *verifiability*: whether the answers to distributed queries are correct. This is a critical concern in any distributed system in which the query processing is shared among the parties. It is certainly a key issue in a φ implementation, since malicious parties on the Internet have every incentive to mask their behavior (and, perhaps, that of others) by computing and/or propagating incorrect results.

We have some initial results in this regard that suggest there is significant room here for innovation. Our work on Proof Sketches [8] can be used to verify the output of distributed aggregation and data-sampling queries. A Proof Sketch allows the parties in the computation to verify that the final result cannot have been maliciously perturbed by more than a small error bound with high probability; when this is not the case, targeted network forensics can be used to identify the node(s) responsible for tampering. The basic technique combines Flajolet-Martin (FM) sketches [6] with compact cryptographic signatures to prevent overcounting; a simple query complement scheme can be used on top of that to prevent undercounting. Using this building block, we are able to support verifiable random sampling as well, which can be used in a broad range of applications for verifiable approximate query processing. To our knowledge, this is the first practical work to tackle verifiable, multi-party query processing in the face of adversaries. It is only a first step, however, and we foresee more work to be done on verifiable distributed queries.

In fact we are convinced that φ opens up a host of new challenges at the boundary between query processing and security, not only in terms of verifiability. Another challenge is to ensure that query specification cannot be used as a weapon: it seems rather easy to specify a distributed query that produces the effect of a Distributed Denial-of-Service (DDoS) attack on an unsuspecting host, for example. This problem has yet to receive attention in the research literature, but it comes up quite naturally if one believes that data integration and peer-to-peer querying are likely to succeed.

3.4 Protocols, Not Systems

The database community has a long tradition of building large, complex systems. This tradition is ill-suited to the design of networked systems. Monolithic systems are typically designed around explicit software modules that export function-call interfaces. By contrast, networked systems are designed around *protocols* that are often implemented in many ways by different parties. The realities of protocol design enforce a methodology that monolithic-system designers do not typically follow. First, and most obviously,

protocols have to be concise, since they determine the messaging overhead in a system, and the main technical documentation burden. Second, protocols have to be simple and clear in their semantics, so that different implementations can interoperate and provide proper behavior. Third, and most importantly, protocols have to be very explicit about exception handling and asynchrony. The endpoints of the protocols are autonomous, and possibly faulty or even malicious. They may not respond at all. They may respond late, or more than once, or out-of-order. These considerations motivate considerable care in thinking through the possible states of the protocol. Protocol design is often done with a mind toward adversarial participants, which is good software-engineering methodology.

Another advantage of designing a system around protocols is that it enables independent innovation at the endpoints. This has been important throughout the history of the Internet, in the development of software for tasks like e-mail, netnews and the Web. The next Grand Challenge for the database community would do well to embrace this approach. Unlike in Operating Systems, the database community has not developed a culture in which most research is done in the context of a single open-source system. Instead, various groups tend to implement (and re-implement) their own software stacks. This culture of competition goes back a long way in the community, and has both merits and demerits. Assuming that it is a reality that needs to be acknowledged, it can be mitigated by building multiple independent software systems that communicate via open shared protocols.

The φ vision pushes database-systems research squarely in this direction. The network is at the heart of the system, as the basic data source, the indexing framework, and the main performance bottleneck. Moreover, the peer-to-peer architecture discourages complex, monolithic, centrally administered software, in favor of simple software that is manageable by end users. The vision of a massive P2P query system with a well motivated application may be the community’s best chance to leverage each other’s systems work.

3.5 Uncertainty, Heterogeneity and Data Reduction

Traditional database users have long known that their data is not perfect: it often contains incorrect, missing, and contradictory information. Various means have been employed to mitigate this problem over the years, including integrity constraint checking, form-based data entry, and a variety of relatively ad-hoc data-auditing and data-cleaning tools.

In our era of extensive data collection and integration, these techniques have been stretched to the limit. As a result, probabilistic reasoning has, perhaps inevitably, made its way into the database community as a first-class research topic. This is a welcome expansion of the community’s agenda, and an ambitious one. The integration of logic, probability, and scalable systems is an enormous challenge that should keep the community busy for some time.

As of now, most of the work in this area is theoretical, and discussion is underway on issues including semantics, data modeling, and query languages. The community has just begun to dig into core issues of managing the statistical cor-

relation models used in modern Machine Learning in the context of a “probabilistic database” [5]. There is a lot of excitement about the database community’s potential to contribute on challenges in the architecture and scalability of probabilistic tasks like learning and inference, but this is as yet a long-term goal. Part of the nascent debate has to do with the diversity of the scenarios being tackled, including the handling of explicitly uncertain observations recorded as data, the integration of “soft” ranking-based systems with “hard” logical query systems, and the acquisition and storage of noisy sensor data [7].

The φ context provides a number of opportunities and constraints in this regard that make it an attractive focus application for investigating probability in queries. Data produced by endpoints in φ will certainly be noisy. Measurements will often be missing. It is unreasonable to expect every single node on the Internet to participate in the system, therefore much traffic will be unobserved. Those nodes that do participate may selectively contribute information for reasons of performance or policies such as privacy. Measurements will also often be erroneous. Some of the data will be events from tools like firewalls that have false positives and negatives. More difficult, some of the data will come from adversarial nodes, who will attempt to “poison” the system with false data for various unseemly purposes.

However, there is quite a bit of structure in Internet traffic that makes it much simpler to handle than almost any other realistic application domain. The scope of semantic heterogeneity in Internet traffic is limited by the fact that all the parties have already standardized on simple data formats in order to interoperate. The entities and relationships described in Internet traces are relatively modest. The most basic data used in network monitoring consists of IP addresses and standard IP packet headers. Moving up the stack, some centralized techniques capture the parameters of standard protocols such as HTTP, SMTP, BGP, etc., as well as simple metadata for Internet Service Providers such as their geographic location. Finally, there are a few dozen endpoint tools that produce simple alerts based on traffic analysis, including firewalls and virus detection tools. (The leading firewall warehouse, DShield, ingests some four-dozen formats worldwide.) It has been shown that effective use can be made of machine-learning methods to combine the results of noisy Boolean alerts across multiple machines [4]. The combination of modest heterogeneity and a realistic application with real data makes φ an attractive setting for exploring more complex probabilistic queries in the context of data that is integrated from many sources.

Moreover, the underlying statistical properties of Internet traffic, while rich, are far better studied than most previously proposed applications of probabilistic databases. Existing (mostly centralized) intrusion-detection systems provide pragmatic statistical models and best practices for identifying “suspicious” Internet traffic. More generally and scientifically, the *Internet Measurement Conference* produces a steady stream of research results on the shifting properties of various aspects of Internet traffic. This modeling energy can be directly harnessed in φ ’s massively distributed setting.

Another opportunity in the φ context is that communica-

tion constraints on the system will highlight linkages between data reduction (a topic the DB community has studied deeply) and statistical inference (largely explored by other communities). Data reduction is a problem of compactly modeling data distributions to minimize storage, access and communication costs. Statistical reasoning typically uses compact models both to better describe data, and to more efficiently run inference algorithms. There has been almost no connection made between our own community’s powerful data-reduction techniques, like sketches [2,9], and statistical inference algorithms like junction trees [21] or loopy belief propagation [20]. This is a rich area for algorithmic investigation that arises naturally in φ , where statistical inference and data reduction are both required.

Finally, network intrusion detection and health monitoring are ongoing areas of application development. It is well understood that the outputs of such a system must be taken with a grain of salt, and there are no expectations of database-style “correctness.” Moreover, there are centralized intrusion-detection systems against which to (a) benchmark the quality of traditional analyses performed in a distributed system, and (b) highlight the additional power of new tools and techniques from the database research community. There are few other realistic settings in which new probabilistic database techniques for structured data can be set in relief against the state of the art.

4. WHY US?

The importance of Internet security is no secret. The problem has been receiving significant attention in both academia and industry for some years now. So it is worthwhile asking what the database community brings to the problem, and whether other research communities have already effectively staked out this territory.

First, it is clear that this problem does not belong to the database community any more than it does to any other research cluster. Malefactors on the Internet do not respect the boundaries between ACM SIGs, and the urgency and persistence of the problem stresses the need for collaboration across the broad computing community. This promotes some positive outcomes for computing in general, and for the database community in particular. It should force meaningful collaborations between researchers in data management, security, networking, machine learning, theory of computation, distributed systems and programming languages, to name a few natural groups. It is not a coincidence that the author list of this paper spans a number of these areas.

Second, it is apparent to all of us that database research does indeed have quite a bit to offer to network-security problems. A brief survey of research on network security highlights a significant need for expertise in distributed query processing and data reduction in particular. One example of this is the DOMINO network-security project at the University of Wisconsin [31], which envisions building something akin to a small PHI infrastructure, but leaves the architecture and algorithms for general distributed query processing for network security largely unspecified. There are many other such examples that either propose very specialized data-processing algorithms for point problems, or that leave query-processing architectures and algorithms to fu-

ture work [12,15,27]. The design of a powerful distributed-query and alert infrastructure for network security is clearly desirable but poorly understood today. Such an infrastructure has to deal meaningfully with many of the challenges we enumerated above. That effort can be significantly accelerated via leadership from the database community, while also engendering collaborations between databases and a number of other fields.

5. THE METACHALLENGE

It is one thing to present a challenge and label it “Grand.” It is quite another to get critical mass within the community working on it. In our own discussions, we puzzle over this metachallenge.

Given interest from a few parties in the community, there are some intriguing recent models that might be worth considering as a group. The first is PlanetLab [22], which was an open effort – seeded with industrial funding – that got a consortium of researchers from a number of institutions to collaborate on a software *platform* that would enable their work. Such a consortium is conceivable in the φ context for designing protocols for distributed querying and mining, and for building reference implementations for some of them. A more traditional academic model is the one used by Project Iris, which focused on DHTs. This effort assembled a group of faculty at five different institutions to apply for significant federal funding, while at the same time starting a new conference (IPTPS). This led to a strong and sustained focus in the research agendas across these groups and others as well, though it did not lead to a great deal of shared code. This model is a bit more difficult to repeat in the current research funding climate. A third model is to go outside the research community for a time and instead push a prototype through the open-source community, perhaps via the Apache Foundation Incubator, or in partnership with a related pre-existing open-source project like SETI@Home. Having done this, the resulting artifact could be used as a platform for researchers.

We eagerly welcome discussion from colleagues on both these metachallenges, and on the φ challenge itself.

6. ACKNOWLEDGMENTS

The authors are grateful to many people for discussions on this topic, and would like to especially thank Kurt Brown, Brent Chun, Nick Feamster, Vern Paxson, Sylvia Ratnasamy, Scott Shenker, Ion Stoica and David Tennenhouse.

7. REFERENCES

- [1] S. Abiteboul, R. Agrawal, P. Bernstein, M. Carey, S. Ceri, B. Croft, D. DeWitt, M. Franklin, H. G. Molina, D. Gawlick, J. Gray, L. Haas, A. Halevy, J. M. Hellerstein, Y. Ioannidis, M. Kersten, M. Pazzani, M. Lesk, D. Maier, J. Naughton, H. Schek, T. Sellis, A. Silberschatz, M. Stonebraker, R. Snodgrass, J. Ullman, G. Weikum, J. Widom, , and S. Zdonik. The Lowell Database Research Self Assessment. <http://research.microsoft.com/~gray/lowell/>, 2003.
- [2] G. Cormode and M. Garofalakis. Sketching streams through the net: Distributed approximate query

- tracking. In *International Conference on Very Large Data Bases (VLDB)*, 2005.
- [3] G. Cormode, R. Keralapura, and J. Ramimirtham. Communication-efficient distributed monitoring of thresholded counts. In *ACM SIGMOD International Conference on Management of Data*, 2006.
- [4] D. Dash, B. Kveton, J. M. Agosta, E. Schooler, J. Chandrashekar, A. Bachrach, and A. Newman. When gossip is good: Distributed probabilistic inference for detection of slow network intrusions. In *Conference on Artificial Intelligence (AAAI)*, 2006.
- [5] A. Deshpande and S. Madden. MauveDB: Supporting model-based user views in database systems. In *ACM SIGMOD International Conference on Management of Data*, 2006.
- [6] P. Flajolet and G. N. Martin. Probabilistic Counting Algorithms for Data Base Applications. *Journal of Computer and System Sciences*, 31, 1985.
- [7] M. Garofalakis and Dan Suciu, eds. Special issue on probabilistic data management. *Bulletin of the Technical Committee on Data Engineering*, 29(1), 2006.
- [8] M. Garofalakis, J. M. Hellerstein, and P. Maniatis. Proof Sketches: Verifiable In-Network Aggregation. In *IEEE International Conference on Data Engineering (ICDE)*, 2007.
- [9] P. B. Gibbons and Y. Matias. Synopsis data structures for massive data sets. In *ACM-SIAM symposium on discrete algorithms (SODA)*, 1999.
- [10] L. Huang, X. Nguyen, M. Garofalakis, J. Hellerstein, A. D. Joseph, M. Jordan, and N. Taft. Communication-efficient online detection of network-wide anomalies. In *IEEE Conference on Computer Communications (INFOCOM)*, 2007.
- [11] R. Huebsch, B. Chun, J. M. Hellerstein, B. T. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A. R. Yumerefendi. The Architecture of PIER: an Internet-Scale Query Processor. In *Conference on Innovative Data Systems Research (CIDR)*, 2005.
- [12] G. Iannaccone, C. Diot, D. McAuley, A. Moore, I. Pratt, and L. Rizzo. The CoMo White Paper. Technical Report IRCTR-04-017, Intel Research, Cambridge, Sept. 2004.
- [13] A. Jain, J. M. Hellerstein, S. Ratnasamy, and D. Wetherall. A wakeup call for internet monitoring systems: The case for distributed triggers. In *Third Workshop on Hot Topics in Networking (HotNets-III)*, 2004.
- [14] R. Jaques. Cost of malware soars to \$166bn in 2004. *vnunet.com*, Feb. 2005. <http://www.vnunet.com/2126635>.
- [15] H.-A. Kim and B. Karp. Autograph: Toward automated, distributed worm signature detection. In *USENIX Security Symposium*, 2004.
- [16] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click modular router. *ACM Trans. Comput. Syst.*, 18(3):263–297, 2000.
- [17] B. T. Loo, T. Condie, M. Garofalakis, D. E. Gay, J. M. Hellerstein, P. Maniatis, R. Ramakrishnan, T. Roscoe, and I. Stoica. Declarative networking with distributed recursive query processing. In *ACM SIGMOD International Conference on Management of Data*, June 2006.
- [18] B. T. Loo, T. Condie, J. M. Hellerstein, P. Maniatis, T. Roscoe, and I. Stoica. Implementing Declarative Overlays. In *ACM Symposium on Operating Systems Principles (SOSP)*, Oct. 2005.
- [19] B. T. Loo, J. M. Hellerstein, I. Stoica, and R. Ramakrishnan. Declarative Routing: Extensible Routing with Declarative Queries. In *ACM SIGCOMM*, Aug. 2005.
- [20] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *15th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.
- [21] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986.
- [22] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A Blueprint for Introducing Disruptive Technology into the Internet. In *First Workshop on Hot Topics in Networking (HotNets-I)*, 2002.
- [23] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *ACM SIGCOMM*, Sept. 2006.
- [24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content Addressable Network. In *ACM SIGCOM*, 2001.
- [25] F. Reiss, M. Garofalakis, and J. M. Hellerstein. Compact histograms for hierarchical identifiers. In *International Conference on Very Large Data Bases (VLDB)*, 2006.
- [26] S. Rhea, D. Geels, T. Roscoe, and J. Kubiawicz. Handling Churn in a DHT. In *USENIX Annual Technical Conference (USENIX)*, 2004.
- [27] V. Sekar, Y. Xie, D. Maltz, M. Reiter, and H. Zhang. Toward a framework for internet forensic analysis. In *Third Workshop on Hot Topics in Networking (HotNets-III)*, 2004.
- [28] S. Staniford, V. Paxson, and N. Weaver. How to Own the internet in your spare time. In *11th USENIX Security Symposium*, 2002.
- [29] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, 2003.

- [30] Y. Xie, V. Sekar, D. A. Maltz, M. K. Reiter, and H. Zhang. Worm origin identification using random moonwalks. In *IEEE Symposium on Security and Privacy (Oakland)*, 2005.
- [31] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the domino overlay system. In *NDSS, Network and Distributed System Security*, 2004.