AFRL-AFOSR-VA-TR-2019-0186

Redefining Analytics for Small High-Performance Computing Clusters

Tim Kraska
BROWN UNIVERSITY

07/15/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01/07/2019	Final	April 1, 2015 - Dec 31, 2018

4. TITLE AND SUBTITLE
Redefining Analytics for Small High-Performance Computing Clusters

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-15-1-0144

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Kraska, Tim

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology, 32 Vassar St, Cambridge, MA 02139
Brown University, 1 Prospect St, Providence, RI 02912

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
USAF, AFRL DUNS 143574726 AF Office of Scientific Research, 875 North Randolph Street, Room 3112, Arlington, VA 22203-1954

10. SPONSOR/MONITOR'S ACRONYM(S)
USAF, AFRL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release.

13. SUPPLEMENTARY NOTES

Transaction processing, RDMA, small high performance cluster, database, high bandwidth network

14. ABSTRACT

Core contributions we made under this grant include (a) the development of the Network-Attached-Memory database architecture, (b) the first scalable RDMA-based transaction protocol, (c) a novel RDMA-based replication protocol, (d) the concept of learned index structures, (e) the first techniques to estimate the impact of Unknown Unknowns on aggregated query results, and (f) novel UDF compilation techniques.
Overall, we were able to address all in the proposal outlined research challenges (RC).
We analyzed the RDMA performance gains (RC I) and developed an RDMA-based storage manager (RC II), we developed modern query execution techniques for UDFs and reinvented the way indexing is done through our learned indexing approach (RC III), we extended our work on data integration in heterogeneous environments (RC IV), we studied the impact of data replication for RDMA-enabled networks (RC V), we significantly advanced the area of UDF and query compilation for complex analytics (RC VI), and developed a novel language to describe ML pipelines (RC VII).

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Tim Kraska
			UU	10	19b. TELEPHONE NUMBER *(Include area code)* 510-926-5856

AFOSR Grant FA9550-15-1-0144 Final Report

Tim Kraska (PI)

Massachusetts Institute of Technology
32 Vassar St., Cambridge, MA 02139
kraska@mit.edu

Brown University
1 Prospect St, Providence RI 02912
e-mail: tim_kraska@brown.edu

1 Status of effort

The grant supported the work of four PhD students (three graduated), contributed to 12 top-tier publications and some of our work (e.g., [17, 16]) opened-up a complete new research directions.

Core contributions we made under this grant include (a) the development of the Network-Attached-Memory database architecture and the according open-source reference implementation NAM-DB [2], (b) a first scalable RDMA-based transaction protocol [17], (c) a novel replication protocol for RDMA [18], (d) the development of the concept of learned index structures [16], (e) the first techniques to estimate the impact of Unknown Unknowns on aggregated query results [5], and (f) novel UDF compilation techniques [6].

Overall, we were able to address all in the proposal outlined research challenges (RC). We analyzed the RDMA performance gains (RC I) in [2] and developed an RDMA-based storage and load-balancing management (RC II) [2, 17], we developed modern query execution techniques for UDFs [6] and reinvented the way indexing is done through our highly acclaimed learned indexing approach [16] (RC III), we extended our work on data integration in heterogeneous environments (RC IV) through our work on unknown unknowns [5], we studied the impact of data replication and high availability for RDMA-enabled networks (RC V) in [18], we significantly advanced the area of UDF and query compilation for complex analytics (RC VI) in [6, 7], and jointly with the DARPA D3M effort we also developed a novel language to describe ML pipelines (RC VII) [1].

2 Key Accomplishments/New Findings:

2.1 Overview

Over the courser of the grant we had the following major accomplishments

(1) NAM-Architecture: The main goal of this grant was to explore how the next generation of high-performance RDMA-capable networks requires to change the architecture of distributed database management systems (DDBMSs). Traditional DDBMSs are designed under the assumption that the network is the bottleneck and thus must be avoided as much as possible. This assumption no longer holds. With Infiniband FDR, the bandwidth available to transfer data across the network is already in the same ballpark as the bandwidth of one memory channel, and the bandwidth is improving fast with upcoming standards. We therefore suggested a new database design, called Network-Attached-Memory DB (NAMDB) [2], which is able to take full advantage of high-bandwidth networks.

(2) Scalable RDMA-based Transaction Protocol: We developed a new snapshot isolation protocol for the NAM-architecture and RDMA-enabled networks and demonstrated that the system can outperform Microsoft's FaRM, which was considered the fastest distributed database system, on the industry-standard TPC-C benchmark [17]. Furthermore, we analytically as well as

empirically showed that distributed transactions — against common wisdom — can scale [17]. This result has far-reaching implications on distributed application design as it simplifies design decisions around co-partitioning schemes, data placement, and handling of inconsistencies.

(3) RDMA-based replication: We demonstrated that existing data replication techniques are sub-optimal for the next-generation of networks as they were designed to minimize network communication between replicas at the cost of incurring more processing redundancy [18]. We further developed a novel *Active-Memory Replication* for NAM-DB that efficiently leverages RDMA to completely eliminate the processing redundancy in replication and achieves significantly higher throughput than alternative techniques.

(4) Learned Index Structures: The quest of finding the best way to find and access data on a remote machine using a single one-sided RDMA operation lead to the idea of building models as indexes. We show, that machine learning can be used to replace index structures with significant space and/or performance benefits [16]. Already shortly after the paper was published on arxiv as a technical report, it received a lot of attention. For example, Steven Sinofsky, the former President of Microsoft's Windows Division, wrote that "This paper blew [his] mind." or Kirk Borne, Principal Data Scientist at Booz Allen, said: "Wow! This could have huge benefits". Since then the core idea behind the paper, to enhance/learn traditional algorithms/data structures, created an entire new line of research and already yielded to a lot of follow up work in the ML and systems community.

(5) UDF compilation techniques: We developed a novel architecture for automatically compiling workflows of user-defined functions (UDFs) [6]. We also propose several optimizations that consider properties of the data, UDFs, and hardware together in order to generate different code on a case-by-case basis. To evaluate our approach, we implemented these techniques in TUPLEWARE [7], a new high-performance distributed analytics system, and our benchmarks show performance improvements of up to three orders of magnitude compared to alternative systems

(6) The Impact of the Unknown Unknown: We developed novel techniques to estimate the impact of the unknown data (a.k.a. unknown unknowns) on simple aggregate queries for common data integration scenarios. The results got published at SIGMOD 2017 [5] and received an ACM TODS "Best of SIGMOD" invitation.

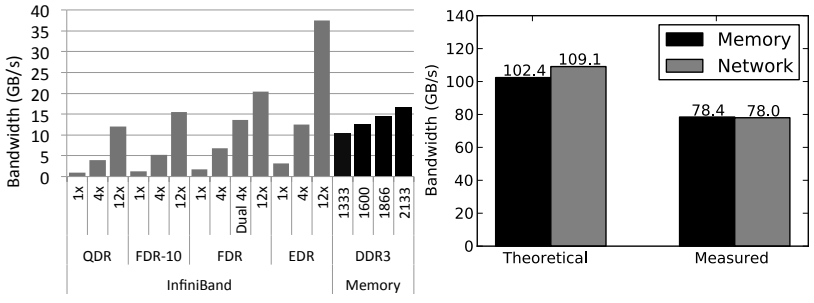
(7) Re-use for Approximate query processing (AQP): We developed a new AQP formulation that can provide low-error approximate results at interactive speeds, even for queries over rare subpopulations [11]. In particular, our formulation treats query results as random variables in order to leverage the ample opportunities for result reuse inherent in interactive data exploration. As part of our approach, we apply a variety of optimization techniques that are based on probability theory, including new query rewrite rules and index structures.

(8) Awards: In part because of the funding of this work, PI Tim Kraska was awarded the 2018 VLDB Early Career Research Contribution Award, the 2017 VMware Systems Research Award, the 2017 Alfred P. Sloan Research Fellow in Computer Science and received the Early Career Research Achievement Award from Brown University. Furthermore, he was hired by MIT from Brown again based on the achievements resulting from this grant.

In the remainder of this section we outline some of the key achievements in more detail.

2.2 The NAM-DB Architecture

Traditional DDBMSs are designed under the assumption that the network is the bottleneck and thus must be avoided as much as possible. This assumption no longer holds true. With Infiniband FDR, the bandwidth available to transfer data across the network is already in the same ballpark as the bandwidth of one memory channel, and the bandwidth is improving fast with upcoming standards. To underline this claim we — to the best of our knowledge — showed for the first time that it is possible with just two dual-port FDR commodity network cards to match the main memory bandwidth of a modern machine (see Figure 1).



(a) Specification (b) Experimental

Figure 1: Memory vs Network Bandwidth: (a) specification, (b) for a Dual-socket Xeon E5v2 server with DD3-1600 and two FDR 4× NICs per socket

Based on this finding, we proposed the network-attached memory (NAM) architecture, which logically decouples compute and storage nodes and uses RDMA for communication between all nodes as shown in Figure 2. The idea is that memory servers

provide a shared distributed memory pool that holds all the data, which can be accessed via RDMA from compute servers that execute transactions. This design already highlights that locality is a tuning parameter. In contrast to traditional architectures which physically co-locate the transaction execution with the storage location from the beginning as much as possible, the NAM architecture separates them. As a result all transactions are by default distributed transactions. However, we allow users to add locality as an optimization like an index. In the following, we give an overview of the tasks of memory servers and compute servers in a NAM architecture.

Memory Servers: In a NAM architecture memory servers hold all data of a database system such as tables, indexes as well as all other state for transaction execution (e.g., logs and metadata). From the transaction execution perspective, memory servers are "dumb" since they provide only memory capacity to compute servers. However, memory servers still have important tasks such as memory management to handle remote memory allocation calls from compute servers as well as garbage collection to ensure that enough free space is always available for compute servers, e.g. to insert new records. Durability of the data stored by memory servers is achieved by using an uninterruptible power supply (UPS). When a power failure occurs, memory servers use the UPS to persist a consistent snapshot to disks. On the other hand, hardware failures are handled through replication as discussed in 2.4.

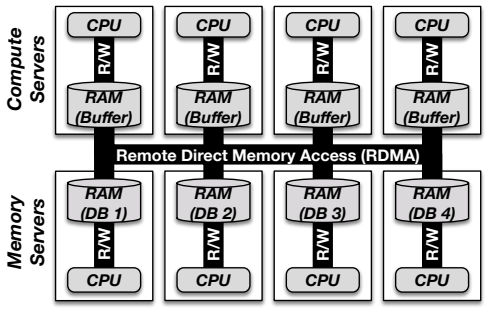


Figure 2: The NAM Architecture

Compute Servers: The main task of compute servers is to execute transactions over the data items stored in the memory servers. This includes finding the storage location of records on memory servers, inserting/ modifying/ deleting records, as well as committing or aborting transactions. Moreover, compute servers are in charge of performing other tasks, which are required to ensure that transaction execution fulfills all ACID properties such as logging as well consistency control. Again, the strict separation of transaction execution in compute servers from managing the transaction state stored in memory servers is what distinguishes our design from traditional distributed database systems. As a result, the performance of the system is independent on the location of the data.

2.3 The End of a Myth: Distributed Transactions Can Scale

It is common wisdom that distributed transactions do not scale. But what if distributed transactions could be made scalable using the next generation of networks and a redesign of distributed databases as proposed in the AFOSR grant? There would no longer be a need for developers to worry about co-partitioning schemes to achieve decent performance. Application development would become easier as data placement would no longer determine how scalable an application is. Hardware provisioning would be simplified as the system administrator can expect a linear scale-out when adding more machines rather than some complex, highly application-specific, sub-linear function.

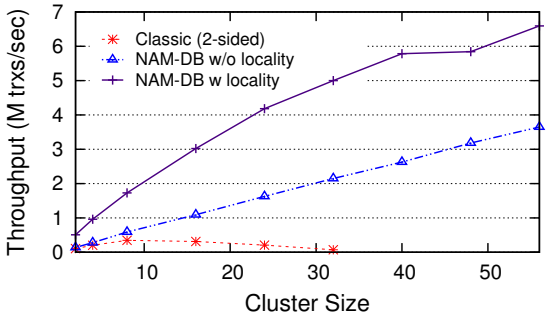


Figure 3: Scalability of NAM-DB

We showed analytically and experimentally that distributed transactions can indeed scale. Building upon the NAM-DB architecture as proposed in the previous year, we developed a new Snapshot Isolation protocol [17]. The key result is shown in Figure 3. It shows the throughput performance of NAM-DB with the new transaction protocol for RMDA with an increasing cluster size. We used the TPC-C benchmark and tested 3 different configurations: (1) our system NAM-DB and protocol without locality (blue). That is, all transactions in the system are completely distributed. (2) NAM-DB with automatic locality optimization (purple). (3) A baseline implementation of a state-of-the-art traditional Snapshot Isolation protocol using 2-sided message-based communication (red). More details about the experimental setup can be found in [17]. The results show that **NAM-DB scales nearly linear** with the number of servers to 3.64 million distributed transactions over 56 machines. This is a stunning result as in this configuration all transactions are distributed. However, if we allow the system to take advantage of locality, we achieve even 6.5 million TPC-C new-order transactions (in TPC-C in the default setting roughly 10% have to be distributed and the new-order transaction makes only up to 45% of the workload). This is 2 million more TPC-C transactions than the current scale-out record by Microsoft FaRM,

which achieves 4.5 million TPC-C transactions over 90 machines with very comparable hardware, the same network technology and using as much locality as possible. It should be noted though, that FaRM implements serializability guarantees, whereas NAM-DB supports snapshot isolation. While for this benchmark, it makes no difference (there is no write-skew), it might be important for other workloads. At the same time though, FaRM never tested their system for larger read queries, for which it should perform particularly bad as it requires a full read-set validation. In order to compare NAM-DB, we also implemented a variant of a distributed SI protocol that follows the classical shared-nothing architecture design with one global master using two-sided RDMA operations for the communication. For this variant, we clearly see that it does not scale with the number of servers used. Even worse, the throughput degrades when using more than 10 machines. This degradation results from the high CPU costs of handling messages. This effect can also be seen by the increased latency.

The result of this sub-project got published at VLDB 2017 and was nominated by Turing-award winner Michael Stonebraker for ACM Computing Review's "Notable Books and Articles in Computing of 2017".

2.4 RDMA-based replication

Highly available database systems rely on data replication to tolerate machine failures. Both classes of existing replication algorithms, active-passive and active-active, were designed in a time when network was the dominant performance bottleneck. In essence, these techniques aim to minimize network communication between replicas at the cost of incurring more processing redundancy; a trade-off that suitably fitted the conventional wisdom of distributed database design. However, the emergence of next-generation networks with high throughput and low latency calls for revisiting these assumptions.

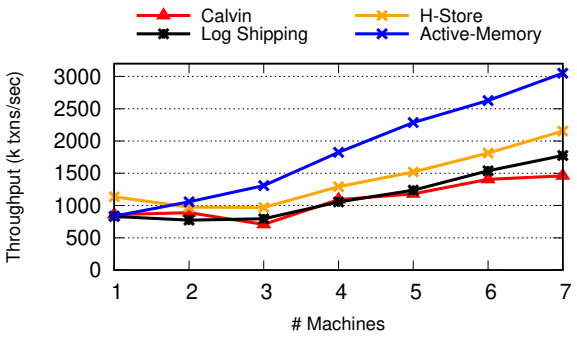


Figure 4: **Scalability** – The throughput of different replication protocols on different cluster sizes. All transactions are single-partition, with each reading and modifying 10 records.

In [18], we first make the case that in modern RDMA-enabled networks, the bottleneck has shifted to CPUs, and therefore the existing network-optimized replication techniques are no longer optimal. We then present *Active-Memory Replication*, a new high availability scheme that efficiently leverages RDMA to completely eliminate the processing redundancy in replication based on the NAM architecture. Using Active-Memory, all replicas dedicate their processing power to executing new transactions, as opposed to performing redundant computation. Active-Memory maintains high availability and correctness in the presence of failures through an efficient RDMA-based undo-logging scheme. Our evaluation against active-passive and active-active schemes shows that Active-Memory is up to a factor of 2 faster than the second-best protocol on RDMA-based networks (see Figure 4.

2.5 Learned Index Structures

One key challenge with one-sided Remote Direct Memory Access (RDMA) operations is, that the physical memory address of the data has to be known before the request to the data is made. This is problematic if we want, for example, to look up the record for a specific customer when we only have his user name. We might know on which remote machine the data is stored, but often not exactly where the data resides in main memory. So far, as part of NAM-DB, we essentially used distributed hash-maps but those leave a lot of main memory unused and did not allow for efficient range requests (e.g., return all records within a specific timeframe). In contrast, more dense storage and index structures, would always require several round-trips to the remote machine and/or require a lot of duplication of the index structure itself. Thus, we started to explore if we can learn a model, that helps determine where the data is stored.

Soon after starting to work on the idea, we not only realized that this is feasible but also that it would provide benefits even within a single machine. As a result, we started to explore the single machine setup first, split across two projects. First, we explored if we can compress a B-Tree index structure using linear regression models. This work was accepted at SIGMOD 2019 [12].

In addition, after a talk at Google in 2017 PI Kraska realized during a meeting, that although this idea cannot compress BTrees, models can entirely replace index structures, not even restricted to BTrees. Given the strong interest of Jeff Dean and others at

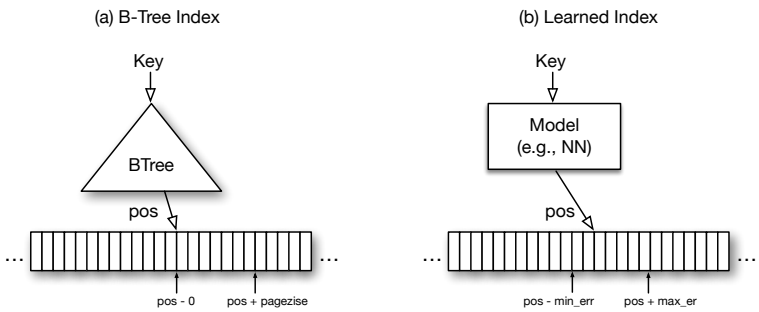


Figure 5: Why B-Tree are models

Google to pursue this idea, PI Kraska started a second project together with Google on the general implications of using machine learning to enhance/replace index structures.

The core idea is based on the observation, that existing index structure are general purpose data structures and (implicitly) assume the worst-case distribution of data without taking advantage of more common patterns prevalent in real world data. For example, if the goal were to build a highly tuned system to store and query fixed-length records with continuous integer keys (e.g., the keys 1 to 100M), one would not use a conventional B-Tree index over the keys since the key itself can be used as an offset, making it an $O(1)$ rather than $O(\log n)$ operation to look-up any key or the beginning of a range of keys. Similarly, the index memory size would be reduced from $O(n)$ to $O(1)$. Maybe surprisingly, the same optimizations are still possible for other data patterns. In other words, knowing the exact data distributions allows for highly optimizing almost any index the database system uses.

Of course, in most real-world use cases the data does not perfectly follow a known pattern and the engineering effort to build specialized solutions for every use case is usually too high. However, we show that machine learning opens up the opportunity to learn a model that reflects the patterns and correlations in the data and thus enable the automatic synthesis of specialized index structures, termed **learned indexes**, with low engineering cost.

We therefore explored the extent to which learned models, including neural networks, can be used to replace traditional index structures from B-Trees to Bloom-Filters. This may seem counter-intuitive because machine learning cannot provide the semantic guarantees we traditionally associate with these indexes, and because the most powerful machine learning models, neural networks, are traditionally thought of as being very expensive to evaluate. Yet, we show that none of these apparent obstacles are as problematic as they might seem. Instead, our proposal to use learned models has the potential for huge benefits, especially on the next generation of hardware.

In terms of semantic guarantees, indexes are already to a large extent learned models - making it surprisingly straightforward to replace them with other types of models, like neural networks. For example, a B-Tree can be considered as a model which takes a key as an input and predicts the position of a data record. To see this, consider a B-Tree index in an analytics in-memory database (i.e., read-only) over the sorted primary key column as shown in Figure 5(a). In this case, the B-Tree provides a mapping from a lookup key into a position inside the sorted array of records with the guarantee that the key of the record at the position is equal or higher than the lookup key. Note that the data has to be sorted to allow for range requests. Also note that this same general concept applies to secondary indexes where the bottom layer would be the list of `<key,pointer>` pairs with the key being the value of the indexed attribute and the pointer a reference to the record. For efficiency reasons, it is common not to index every single key of the sorted records, rather only the key of every n-th record, i.e., the first key of a page.[1] This helps to significantly reduce the number of keys the index has to store without any significant performance penalty. Thus, the B-Tree is a model, in ML terminology a regression tree: it maps a key to a position with a min- and max-error (a min-error of 0 and a max-error of the page-size) and the guarantee that the key can be found in that region if it exists. Consequently, we can replace the index with other types of machine learning models, including deep learning models, as long as they are also able to provide similar strong guarantees about the min- and max-error (see Figure 5(b)).

Similarly, a Bloom-Filter is a binary classifier, which based on a key predicts if a key exists in a set or not. Obviously, there

[1]Here, we assume logical paging over the sorted array, not physical pages which are located in different memory regions. The latter is particularly required for inserts and/or disk-based systems; we will address real paging later in the paper. Also, we assume fixed-length records, with points to overflow regions for variable-length records.

exist subtle but important differences. For example, a Bloom-Filter can have false positives but not false negatives. However, as we show in our paper, it is possible to address these differences through novellearning techniques and/or simple auxiliary data structures.

In terms of performance, we observe that every CPU already has powerful SIMD capabilities and we speculate that many laptops and mobile phones will soon have a Graphics Processing Unit (GPU) or Tensor Processing Unit (TPU). It is also reasonable to speculate that CPU-SIMD/GPU/TPUs will be increasingly powerful as it is much easier to scale the restricted set of (parallel) math operations used by neural nets than a general purpose instruction set. As a result, the high cost to execute a neural net might actually be negligible in the future. For instance, both Nvidia and Google's TPUs are already able to perform thousands if not tens of thousands of neural-net operations in a single cycle. Furthermore, it was stated by NVIDIA's CEO that GPUs will improve $1000\times$ in performance by 2025, whereas Moore's law for CPU essentially is dead. By replacing branch-heavy index structures with neural networks, databases can benefit from these hardware trends.

It is important to note that we do not argue to completely replace traditional index structures with learned index structures. Rather, we outline a novel approach to build indexes, which complements existing work and, arguably, opens up an entirely new research direction for a decades-old field. While our focus is on read-only analytical workloads, we also sketch how the idea could be extended to speed-up indexes for write-heavy workloads. Our initial results show, that by using neural nets we are able to outperform cache-optimized B-Trees by up to 70% in speed while saving an order-of-magnitude in memory over several real-world data sets. More importantly though, we believe that the idea of replacing core components of a data management system through learned models has far-reaching implications for future systems designs and that this work just provides a glimpse of what might be possible.

This work was published at SIGMOD 2018 [16] and received a lot of attention on social media. Furthermore, it started a whole new line of research on learned algorithms and data structure, and also lead to our new research project SageDB [15] and the creation of the DSAIL lab at MIT. Finally, I am in close contact with several industry leaders from Oracle to Microsoft about how to transition the technology into practice.

2.6 Estimating the Impact of Unknown Unknownson Aggregate Query Results

As outlined in *Research Challenge IV: Legacy Systems and Data Integration* of our AFOSR proposal, the usefulness of SHPC also depends on the capability to integrate with other data sources and make the data available quickly. While there exist sophisticated systems from industry and academia alike to assist data scientists in the process of data integration, two fundamental questions remain unanswered: (1) do the data sources cover the complete data set of interest and (2) what is the impact of any unknown (i.e., unobserved) data on query results?

In this sub-project, we develop techniques to estimate the impact of the unknown data on aggregate queries of the form `SELECT AGGREGATE(attr) FROM table WHERE predicate`. We assume a simple data integration scenario where several domain-related data sources are integrated into one database, preserving the lineage information for each data item or record. Naturally, these data sources overlap with each other, but even when put together they might not be complete. Estimating the impact of the unknown data (data items that are not observed in any data source) is particularly difficult as we neither know how many unique data items are missing nor their values; thus, we deal with **unknown unknowns**. This characteristic distinguishes our work from what is generally known as *missing data*, or *known unknowns*, estimation in Statistics, which tries to estimate the value of unknown (missing) attributes for known records. At a first glance, it may seem impossible to estimate the impact of *unknown unknowns*; however, for a large class of data integration scenarios, the analysis of overlap of multiple data sources makes it feasible.

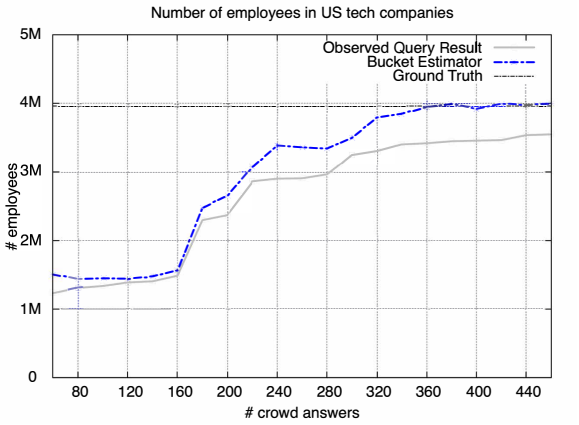


Figure 6: Employees in the U.S. tech sector

To demonstrate the impact of *unknown unknowns*, we pose a simple aggregate query to calculate the number of all employees in the U.S. tech industry, `SELECT SUM(employees) FROM us_tech_companies`, over a crowdsourced data set. The data was manually cleaned before processing (e.g., entity resolution, removal of partial answers). Figure 6 shows the result. The dotted gray line represents the ground truth (i.e., the total number of employees in the U.S. tech sector) for the query, whereas the

solid line shows the result of the observed SUM query over time with the increasing number of received crowd-answers. The gap between the observed and the ground truth is due to the impact of the *unknown unknowns*, which gets smaller at a diminishing rate as more crowd-answers arrive.

While the experiment was conducted in the context of crowdsourcing, the same behavior can be observed with other types of data sources. For instance, suppose a user searches the Internet to create a list of all solar energy companies in the U.S. The first few web pages will provide the greatest benefit (i.e., more new solar companies), while after a dozen web pages the benefit of adding another web page diminishes as the likelihood of duplicates increases. The rate of increasing overlap of data sources is indicative of the completeness of the data set.

This work is a first step towards developing techniques to estimate the impact of the *unknown unknowns* on query results. Our focus is on simple aggregate queries, especially *SUM*-aggregates, but we also touch upon other aggregations like *COUNT*, *AVG*, *MIN*, and *MAX*. We design techniques that can deal with the peculiarities of the data-integration scenarios discussed before, such as uneven contributions from different sources (bias of data sources). In summary, this sub-project made following key contributions:

- We formalize the problem of estimating the impact of *unknown unknowns* on query results and describe why existing techniques for species estimation and missing data estimation are not sufficient.
- We develop techniques to estimate the impact of the *unknown unknowns* on aggregate query results.
- We derive a first upper bound for *SUM*-aggregate queries.
- We examine the effectiveness of our techniques via experiments using both real and synthetic data sets. For example, Figure 6 shows the results for our bucket estimator, which is almost perfectly able to correct the query result to the ground truth value.

The results of this work got published at SIGMOD 2016 [5] and received an invite from the ACM Transactions on Database Systems journal as a special "Best of SIGMOD" article. Furthermore, based on the results we also recently developed estimates for the quality of data [4].

2.7 Re-use for Approximate query processing (AQP)

One of the key challenges in introducing new hardware and software is the integration with legacy systems, as outlined in *Research Challenge IV: Legacy Systems and Data Integration* of our grant proposal. This is particularly prevalent for data-analytic systems as most of the data is and will remain outside of the new system for various reasons. To overcome this problem we proposed IDEA, a first Interactive Data Exploration Accelerator [8]. IDEA allows the seamless integration as it supports a connect and explore paradigm: the user can connect to any data source and the system in the background automatically starts to intelligently sample and pre-process data so that any queries and tasks can be performed over the samples with strong guarantees over the results.

We use Approximate Query Processing (AQP) and Online Aggregation techniques to return query results at "human speed" by approximating query answers. However, existing AQP techniques start to break down when confronted with ad hoc queries that target the tails of the distribution. Since the exploration of rare subpopulations (e.g., high-value customers, anomalous sensor readings) often leads to the most significant insights, AQP/online aggregation falls short when confronted with these types of workloads.

We therefore explored a new AQP formulation that can provide low-error approximate results at interactive speeds, even for queries over rare subpopulations. In particular, our formulation treats query results as random variables in order to leverage the ample opportunities for result reuse inherent in interactive data exploration. As part of our approach, we apply a variety of optimization techniques that are based on probability theory, including new query rewrite rules and index structures. We implemented these techniques in a prototype system called IDEA [8] and show that they can achieve interactivity where alternative approaches cannot.

The results of this work got published at VLDB 2017 [11], IEEE Data Eng. Bull. [9], at the HILDA workshop at SIGMOD 2016 [8, 10], and as an invited paper to SIGMOD 2017 [14].

3 Publications

[1] C. Binnig, B. Buratti, Y. Chung, C. Cousins, T. Kraska, Z. Shang, E. Upfal, R. C. Zeleznik, and E. Zgraggen. Towards interactive curation & automatic tuning of ML pipelines. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning, DEEM@SIGMOD 2018, Houston, TX, USA, June 15, 2018*, pages 1:1–1:4, 2018.

[2] C. Binnig, A. Crotty, A. Galakatos, T. Kraska, and E. Zamanian. The end of slow networks: It's time for a redesign. *PVLDB*, 9(7):528–539, 2016.

[3] M. J. Cafarella, I. F. Ilyas, M. Kornacker, T. Kraska, and C. Ré. Dark data: Are we solving the right problems? In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 1444–1445, 2016.

[4] Y. Chung, S. Krishnan, and T. Kraska. A data quality metric (DQM): how to estimate the number of undetected errors in data sets. *PVLDB*, 10(10):1094–1105, 2017.

[5] Y. Chung, M. L. Mortensen, C. Binnig, and T. Kraska. Estimating the impact of unknown unknowns on aggregate query results. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 861–876, 2016.

[6] A. Crotty, A. Galakatos, K. Dursun, T. Kraska, C. Binnig, U. Çetintemel, and S. Zdonik. An architecture for compiling udf-centric workflows. *PVLDB*, 8(12):1466–1477, 2015.

[7] A. Crotty, A. Galakatos, K. Dursun, T. Kraska, U. Çetintemel, and S. B. Zdonik. Tupleware: "big" data, big analytics, small clusters. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*, 2015.

[8] A. Crotty, A. Galakatos, E. Zgraggen, C. Binnig, and T. Kraska. The case for interactive data exploration accelerators (ideas). In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics, HILDA@SIGMOD 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, page 11, 2016.

[9] P. Eichmann, E. Zgraggen, Z. Zhao, C. Binnig, and T. Kraska. Towards a benchmark for interactive data exploration. *IEEE Data Eng. Bull.*, 39(4):50–61, 2016.

[10] M. El-Hindi, Z. Zhao, C. Binnig, and T. Kraska. Vistrees: fast indexes for interactive data exploration. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics, HILDA@SIGMOD 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, page 5, 2016.

[11] A. Galakatos, A. Crotty, E. Zgraggen, C. Binnig, and T. Kraska. Revisiting reuse for approximate query processing. *PVLDB*, 10(10):1142–1153, 2017.

[12] A. Galakatos, M. Markovitch, C. Binnig, R. Fonseca, and T. Kraska. Fiting-tree: A data-aware index structure. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019.*, pages 1189–1206, 2019.

[13] B. Howe, M. J. Franklin, L. M. Haas, T. Kraska, and J. D. Ullman. Data science education: We're missing the boat, again. In *33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017*, pages 1473–1474, 2017.

[14] T. Kraska. Approximate query processing for interactive data science. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, page 525, 2017.

[15] T. Kraska, M. Alizadeh, A. Beutel, E. H. Chi, A. Kristo, G. Leclerc, S. Madden, H. Mao, and V. Nathan. Sagedb: A learned database system. In *CIDR 2019, 9th Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 13-16, 2019, Online Proceedings*, 2019.

[16] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 489–504, 2018.

[17] E. Zamanian, C. Binnig, T. Kraska, and T. Harris. The end of a myth: Distributed transaction can scale. *PVLDB*, 10(6):685–696, 2017.

[18] E. Zamanian, X. Yu, M. Stonebraker, and T. K. (MIT). Rethinking database high availability with rdma networks. *PVLDB*, 2019.

4 Students supported

- Yeounoh Chung, PhD student
- Andrew Crotty, PhD student
- Alexander Galakatos, PhD student
- Erfan Zamanian Dolati, PhD student

5 Interactions/Transitions

5.1 Selected Invited Colloquium Speaker / Special Seminars

- **The End Of Systems As We Know Them**, TTI Vanguard 12/2018, Microsoft Senior Management Meeting (including CEO Satya Nadella) 11/2018
- **Learned Index Structures**, ETH Zurich 05/2019, TU Munich 05/2019, University of Wisconsin 05/2019, New England Network and Systems Day (NENS) (**keynote**) 04/2019, UC Berkeley 03/2019, Oracle 12/2018, Google 03/2019, MSR Faculty Summit 07/2018, SIGMOD 06/18, Data Science Summit 05/2018, XLDB 04/2018,

  Cloudera 01/2018, and Microsoft 01/2018, Facebook 01/2018, Buffalo University 12/2017
- **Interactive Data Science**, SAP 06/2018, Tel Aviv University 06/2018, Aalborg University 08/2016, University of Massachusetts 10/2016, IIT Delhi 11/2016, VLDB (**award talk**) 08/2018
- **Quantifying the Uncertainty in Data Exploration**, University of Washington, Seattle, WA 01/2017
- **The End of a Myth: Distributed Transactions Can Scale**, Future Cloud 04/2018

5.2 Selected Invited conference/workshop/panel speaker

- **How not to fail (keynote)**, VLDB Ph.D. workshop, Delhi, India, 08/2016 (Kraska)
- **Dark Data: Are we solving the right problems? (panel)**, IEEE International Conference on Data Engineering (ICDE), Finland, 05/2016
- **The End of Slow Networks: It's Time for a Redesign** VLDB, Delhi, India, 08/2016 (Zamanian)
- **Estimating the Impact of Unknown Unknowns on Aggregate Query Result** SIGMOD Conference, San Francisco, CA, 06/2016 (Chung)
- **Machine learning and storage: Why is our community not at the table? (panel)**, Conference on Innovative Data Systems Research (CIDR), Santa Cruz, CA, 01/2017 (Kraska)
- **Data Sharing and Integration: Infrastructure and Best Practices (panel)** invited on behalf of Governor Malley, Mayor Murray of Seattle, and the University of Washington), Big Data + Human Services Workshop, Seattle, WA, 01/2017 (Kraska)
- **Data science education: We are missing the boat, again (panel)**, IEEE International Conference on Data Engineering (ICDE), San Diego, CA, 04/2017 (Kraska) [13] [3]

- **Approximate Query Processing for Interactive Data Science (keynote)** SIGMOD, Chicago, IL, 05/2017 (Kraska)
- **Learned Index Structures (kenynote)**, O'Reilly AI 05/2018
- **Learned Index Structures (kenynote)**, AIDM@SIGMOD workshop 06/2018
- **The End Of Systems As We Know Them (keynote)**, O'Reilly AI 06/2019

6 Honors Received During the Period of the Award

- VLDB Early Career Research Contribution Award 2018 (Kraska)
- VMware Systems Research Award 2018 (Kraska)
- Alfred P. Sloan Research Fellow in Computer Science, 2017 (Kraska)
- Early Career Research Achievement Award, Brown University, 2017 (Kraska)
- VMware Early Career Faculty Grant, 2017 (Kraska)
- ACM TODS - Best of SIGMOD invitation 2016 for our work on "Estimating the Impact of Unknown Unknowns on Aggregate Query Result"
- ACM Computing Review's "Notable Books and Articles in Computing of 2017" nomination for our work on "The end of a myth: Distributed transaction can scale"