Small Data
(Panel)

Oliver Kennedy^{b}, D. Richard Hipp^{r}, Stratos Idreos^{h}, Amélie Marian^{r},
Arnab Nandi^{o}, Carmela Troncoso^{i}, Eugene Wu^{c}
b: University at Buffalo, r: Hipp, Wyrick & Company, Inc h: Harvard, r: Rutgers,
o: Ohio State University, i: IMDEA Software Institute, c: Columbia University,
okennedy@buffalo.edu, drh@sqlite.org, stratos@seas.harvard.edu, amelie@cs.rutgers.edu
arnab@cse.ohio-state.edu, carmela.troncoso@imdea.org, ew2493@columbia.edu

***Abstract*—Data is becoming increasingly personal. Individuals regularly interact with a wide variety of structured data, from SQLite databases on phones, to HR spreadsheets, to personal sensors, to open government data appearing in news articles. Although these workloads are important, many of the classical challenges associated with scale and Big Data do not apply. This panel brings together experts in a variety of fields to explore the new opportunities and challenges presented by "Small Data"**

I. Overview

Over a decade ago, challenges to assumptions like "Distributed systems failures are outliers", "We can't collect everything", and "There isn't enough data to distinguish signal from noise" led us into the big data era. Now, fundamental assumptions are changing again. Smart devices are making data more personal. Intelligence is moving closer to the edge with low-cost embedded computing platforms. Tools like D3 are making interactive visualizations a key part of news reporting. Interfaces like Wolfram Alpha and Siri are putting complex question-answering within easy reach. In short, we are transitioning to an era where where the data management bottleneck is personal and per-device interactions, rather than scalability — an era of "Small Data". This panel will facilitate a discussion of small data and encourage participants to challenge long-held data management assumptions. After brief 2-3 minute self-introductions, this panel will encourage audience engagement through an open debate and discussion format. Topics for discussion will include: (1) What is small data and why should the database community care? (2) How do human factors affect data management systems and how data is accessed? (3) As edge computing devices like smart sensors, embedded linux, phones, and watches become pervasive, what bottlenecks will DBMSes have to contend with? (4) Is SQL the right language for a landscape dominated by imperative programming? (5) What tools are required to help individuals leverage open public data? (6) How should new small data technologies be evaluated? (7) What resources are available for new research on small data?

II. Moderator and Panel

Oliver Kennedy (Moderator) is an assistant professor at the University at Buffalo, working on uncertain data, database usability, query optimization, and data structures. Oliver's work includes DBToaster (as seen in the best of VLDB 2012 issue of VLDBJ), Mimir (a user-friendly probabilistic ETL tool), and PocketData (an embedded database benchmark based on usage patterns from Android smartphones in the wild).

In his view, data becomes a vastly different beast at smaller scales. At small scales, the law of large numbers is inapplicable and there is insufficient noise to absorb all of the outliers. At small scales, users feel comparatively comfortable making and relying on sweeping assumptions about the data. At small scales, users often use more comfortable imperative, rather than declarative programming models. And at small scales, rapid-fire interactive data exploration is the norm.

D. Richard Hipp got his PhD at Duke University in 1992, and is the original author and principle maintainer for the SQLite database engine, the most widely used database engine in the world. Richard is also the founder and a co-owner of Hipp, Wyrick & Company, Inc., a North Carolina company that provides advanced software design and implementation services for an international clientele.

In his view, a data management system for big data typically consists of thousands of cooperating nodes. The system needs to deal with occasional node outages and network failures, but benefits from having nodes that are in a data center at the core of the network, with well-conditioned power and lavishly endowed with memory and CPU cycles, and lovingly attended by a professional IT staff. Small data systems, in contrast, run on millions or billions of autonomous devices operating at the edge of the network. RAM and CPU cycles are comparitively scarce, power comes from batteries or other sources subject to frequent interruption, and no human experts are at hand to monitor performance, trouble-shoot problems, or even to keep the system up-to-date with the latest patches.

These differences have important design and engineering implications. Data management systems for small data must be efficient due to resource limitations, and simple in order to promote the robustness needed for long-term, reliable, and unattended operation. Concurrency is often reduced or omitted for the sake of simplicity and because it is not normally needed. Adaptive techniques which adjust processing based on the quantity or "shape" of the stored data are avoided so that systems have the same performance characteristics after deployment as they did in the lab. Developing a data management system for small data requires a different mindset. Instead of constantly thinking "How can I run this operation in parallel?", developers of small data systems focus on questions like "How can I make this operation draw less energy from the battery?".

Stratos Idreos is an assistant professor of Computer Science at Harvard University where he leads DASlab, the Data Systems

Laboratory@Harvard SEAS. Stratos' work emphasizes making it easy to design efficient data systems as applications and hardware keep evolving and on ease-of-use for non-experts. Stratos is the recipient of numerous awards including the 2011 ACM SIGMOD Jim Gray Doctoral Dissertation award, a VLDB Challenges and Visions best paper award, the NSF CAREER award, and an IEEE TCDE Early Career award from the IEEE Technical Committee on Data Engineering for his work on adaptive data systems.

In his view, not all data sets are huge. Still all analytics tasks are best handled by efficient systems. Using a system that is meant for terabytes of data and hundreds of machine nodes to handle the typical case of a small dataset quickly becomes an expensive and complex process. In most cases, simply using tools such as python, excel or even perl, awk, or similar, provides a quicker and cheaper solution. It is not the most efficient, though, and it become a bottleneck if data or functionality scale. What if there was a way to easily create tailored systems that can handle small data sets efficiently, for example with the ease of using a tool such as python and with the efficiency of a full blown tailored system?

Amélie Marian is an Associate Professor in the Computer Science Department at Rutgers University, where she leads the DigitalSelf project, which aims at providing users with tools to regain control of and exploit their digital data traces. Her research interests include personal information management, semi-structured data processing and web data management.

Arnab Nandi is an assistant professor at Ohio state. Arnab's research is in the area of database systems, focusing on exploiting user behavior to address challenges in large-scale data analytics and interactive data exploration. Arnab is a founder of The STEAM Factory, a collaborative interdisciplinary research and public outreach initiative, and faculty director of the OHI/O Hackathon Program. Arnab is a recipient of the NSF CAREER Award, a Google Faculty Research award, an IEEE TCDE Early Career award, and the Ohio State College of Engineering Lumley Research award.

In his view, one of the critical advantages with small data is the increased "human attention-per-tuple" ratio. With reduced volume, variety, and velocity of data, the data practitioner (the "user") can now pay attention to most, if not all of the data. This entails a lower cognitive overload for the user, and hence, allows the user to invest more time and curation effort into preparing and analyzing the data, possibly reducing room for error. Further, given the limited amount of data involved, the user can now iterate through the data faster, and hence tighten the $questions \rightarrow insights \rightarrow questions$ loop.

These new observations lead to new opportunities in building data management systems. Drawing insights and deductions from small data is easier due to lower cognitive overload. However, from a correctness standpoint, getting to the smaller data (i.e., data reduction) will need to be done carefully, since small data can often involve a non-representative sample of a larger dataset. Second, this would will allow for richer questions (DSLs, query models) and richer answers (data models, representation schemes). Finally, from a performance standpoint, explicitly articulating "small" as a requirement would allow us to consider time-boundedness as a design consideration across the stack, e.g., even at the query execution layer. This would enable us to think about "feedback-first" databases, where we not only think about the Query and the Result, but also about the Feedback provided in realtime(i.e., $\sim O(1)$ responses) to the user as they are formulating their query / playing with the data to get a sense of it.

Carmela Troncoso is a researcher at the IMDEA Software Institute where she leads the research line on privacy enhancing technologies. Her research focuses on developing systematic means to analyze and design robust privacy-preserving systems. She has over 30 articles in top Security and Privacy venues, and is part of the board of the Privacy Enhancing Technologies Symposium, that she will chair in 2018.

In her view, the raise of powerful personal devices that allow to produce, process, and collect small amounts of data opens the door to the development of highly decentralized applications. This enables the construction of applications with deep implications in security and privacy. On the one hand decentralization can be very beneficial from a privacy perspective, since it enables users to keep better control over their data or even not disclose them to centralized services. On the opposite side, providing individuals with capability to collect and generate data has fostered the appearance of crowdsourcing applications in which users can contribute their data enabling the collection of large datasets at low cost, that entail high privacy risks for users.These scenarios challenge the trust assumptions under which we design and develop current data-driven service, since it becomes hard to verify the trustworthiness of participating entities, and it is not guaranteed that any of them will have a global vision of the system to coordinate and execute security-oriented mechanisms.

Changes in the edge devices capabilities and its impact on the trust model require rethinking the design of data management systems to achieve different security or privacy properties. Key questions that need to be studied are: what data should be collected to enable privacy? How should this data be collected so that the meta-data associated does not entail a privacy breach? how do we ensure data authenticity and integrity? how do we establish trust within the different nodes in the system? and trust in the data management process?

Eugene Wu is an assistant professor of Computer Science at Columbia University focusing on accelerating the democratization of data. His interests include algorithms to explain data analysis results, data cleaning and preparation, and data visualization management systems.

In his view, data analysis is a process of "run analysis, look at results, think, repeat". Data processing, part of "run analysis", has traditionally been the "big" bottleneck, however Moore's law has effectively rendered the majority of data sets "small" and fast to analyze. In this setting, other tasks—looking at and interpreting results, thinking about and expressing next steps, even accessing and collecting useful data—become dominant costs that must be addressed in a way that accounts for the user goals. In addition, for this setting, it's not clear whether a monolithic DBMS, or a disparate collection of utilities, or hybrid solution is most useful.