Highly Scalable Data Processing Framework for Pervasive Computing Applications

Janne Riihijärvi and Petri Mähönen
Institute for Networked Systems, RWTH Aachen University
Kackertstrasse 9, D-52072 Aachen, Germany
Email: {jar, pma}@inets.rwth-aachen.de

***Abstract*—One of the key problems in pervasive computing is enabling the collective processing of sensor data obtained from mobile devices such as smartphones. In this demonstration we present a highly scalable storage and processing framework for pervasive computing applications, enabling various estimation problems to be solved from massive data sets, consisting of measurements from millions of nodes or more. The key to achieving such scalability is the use of linear or sublinear time processing algorithms emerging from statistical and machine learning communities. We focus specifically on *spatial* and *spatio-temporal* estimation problems in the demonstration, such as prediction of sensor readings, user densities, or wireless network usage in regions for which direct measurements are not available.**

***Keywords*-Pervasive computing; massive data sets; sublinear methods; fixed rank kriging**

I. Introduction

The rapid proliferation of smartphones is presenting the pervasive computing community with an unprecedented data processing challenge. Together with other mobile devices smartphones are enabling massive amounts of diverse types of sensor data to be collected, annotated with a geographical location and time of acquisition. Processing this data for studying, for example, user mobility dynamics, contact patterns, or the radio coverage of various wireless systems presents a massive spatio-temporal estimation problem. Existing platforms (see, for example, [1], [2] and references therein) have typically focused on the data gathering and annotation problem, without specifically focusing on the processing challenge.

In this demonstration we will present a prototype implementation of a data storage and processing framework specifically tailored for pervasive computing applications, with scalability properties needed for dealing with the emerging massive data sets. Our prototype builds on our earlier work on radio coverage estimation [3], which has been substantially extended to support additional sensor modalities and data processing algorithms. We focus specifically on spatio-temporal estimation problems in the demonstration, basing our work on recently developed *fixed rank* spatial and spatio-temporal estimation methods [4], [5], [6] as well as sublinear machine learning techniques [7], [8]. We specifically leave out the problems related to data acquisition from the demonstration, since those are already well addressed in the literature in frameworks such as [1], and for specific sensor modalities are even already being dealt with in standardization bodies [9], [10].

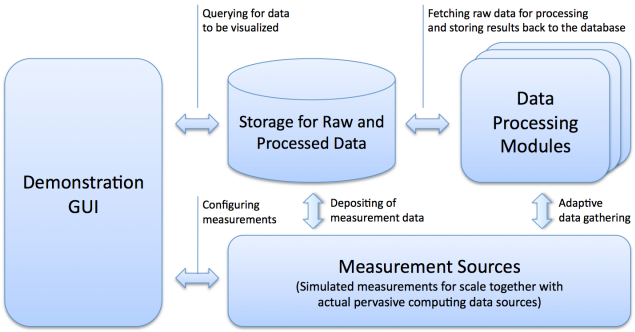

Figure 1. The high level functional architecture of the demonstrated data processing framework.

In the following sections we first give an overview of the design and overall architecture of the demonstrated framework, and then discuss the implemented data processing algorithms and related demonstrated functionalities in more detail.

II. System Architecture and Design

The high level architecture of the developed data processing framework is illustrated in Figure 1. Different data sources deposit measurement results into a logically centralized storage service, from which they are accessed by different data processing modules. These are run either periodically, or in an event-driven fashion based on changes or additions of new sensor readings into the database. The results from the data processing modules are also stored back into the storage service, from which they can be accessed by other data processing modules for subsequent processing or refinement of results, or by the demonstration GUI for visualization. The data processing modules can also distribute measurement plans or schedules to data sources, for example to improve the accuracy of results in regions in which too small amount of data was originally available.

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

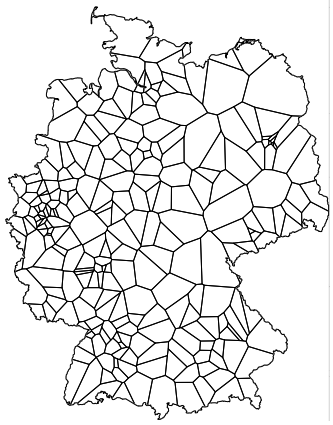

Figure 2. An example of a spatial tessellation adapted to the distribution of measurement locations used for indexing sensed data for storage.

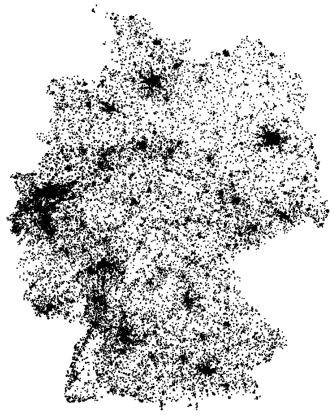

Figure 3. Simulated measurement device population following the population distribution of Germany.

It should be emphasized that the architecture shown in Figure 1 is, especially regarding the storage service, only *logically* centralized, with the individual architectural elements typically being implemented in distributed fashion to avoid bottlenecks and to increase reliability. For example, as a basis of the storage service we have worked with both Hadoop [11] and more recently developed HyperDex no-SQL storage system [12]. Both of these offer distributed, highly scalable and fault tolerant platforms for storing, querying and processing data. In order to utilize such storage platforms as a part of our framework, a number of novel design choices have been incorporated into the implementation. For example, to support spatial indexing of data within the storage platforms, spatial tessellations such as shown in Figure 2 are used to derive indexes for measurement locations in a scalable manner. These design decisions will be further detailed in a poster accompanying the demonstration, enabling the attendees to have better insight into the implementation challenges in such systems.

III. Technical Requirements and Functionality

In this section we outline in more detail the technical details and the functionality of the proposed demonstration.

A. Demonstration Setup and Flow

The demonstration will support both *local* as well as *remote* mode. In the former, all the elements of the functional architecture are run on a single laptop as separate processes communicating through Websocket-based interfaces. A variety of prerecorded measurement traces as well as dynamically generated simulated traces will be available as data sources. Through the GUI attendees can run different processing and estimation algorithms — discussed in more detail below — on the corresponding data sets, and study the results compared against the ground truth. Figure 3 illustrates one of the simulated data source types, corresponding to a percentage of smartphone users in Germany. On-the-fly simulation of data sources is supported in order to allow the attendees to study the impact of the amount and quality of the available measurement data on the accuracy and computational performance of the implemented storage and processing solutions.

In the remote mode most parts of the storage framework together with the data processing modules are run in a distributed fashion on a number of dedicated computational servers, with the demonstration laptop only being used for the user interface, and for providing local storage area for the results to be visualized. The same interfaces as in the local mode are still used, and the actual codebase used in both demonstration modes is identical. The remote mode enables demonstration of the distributed storage and processing aspects, while the local mode is provided for robustness, making the demonstration setup independent of working Internet access.

B. Examples of Implemented Data Processing Algorithms

As discussed in the introduction, our focus is especially on algorithms for spatio-temporal estimation problems. Figure 4 illustrates such a problem in which values of spatially continuous phenomenon such as temperature, user density, or received signal strength has to be estimated for a given region based on a collection of samples. The figure shows the ground truth data set, as well as optimal reconstructions using techniques from spatial statistics [13] for different densities of sensor nodes. The fixed rank estimation algorithms [5] implemented in the demonstrated framework enable such estimates to be made for massive data sets, having only linear time computational complexity in the number of measurements available. For each of the estimation algorithms, the user interface enables studying the accuracy of the obtained estimates, the computational time

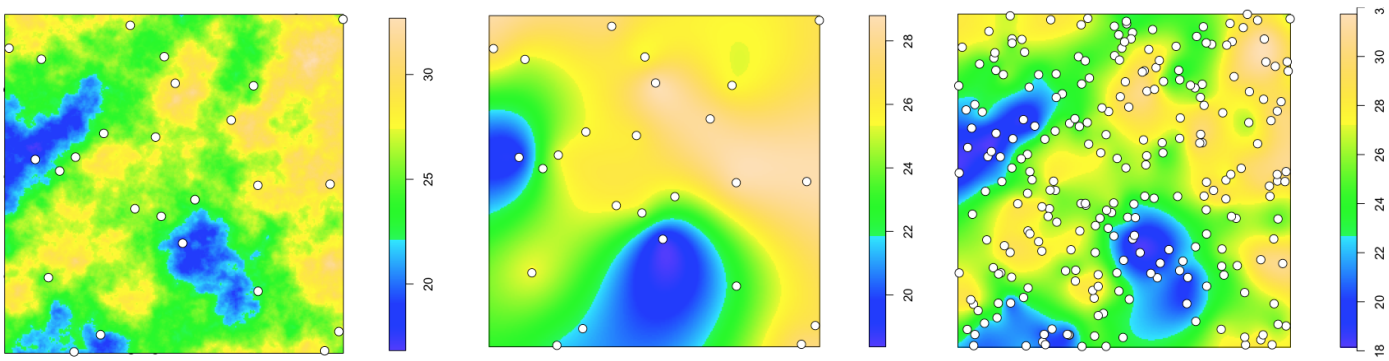

Figure 4. Convergence of spatial sampling and estimation, with panel on the left showing the ground truth, middle panel showing the optimal reconstruction based on 25 sensor readings, and panel on the right showing a more accurate reconstruction with 250 sensors.

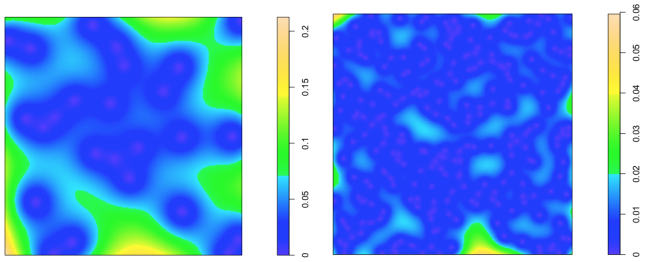

Figure 5. Formal variance estimates of the spatial predictions shown in the middle and right panels of Figure 4.

involved, and how these depend on the properties of the underlying data set, such as the number of simulated mobile devices carrying out the measurements. Further, many of the algorithms are also capable of estimating the reliability of their results, and these estimates together with their reliability can be explored through the demonstration GUI. For example, in Figure 5 the estimated prediction errors are shown for the results given in Figure 4. Such estimates can be used to drive adaptive measurement routines, and to improve the battery life on mobile terminals carrying out the measurements.

Acknowledgment

We thank RWTH Aachen University and the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) for providing financial support for this work through the UMIC research centre.

References

[1] J. Kukkonen, E. Lagerspetz, P. Nurmi, and M. Andersson, "Betelgeuse: A platform for gathering and processing situational data," *Pervasive Computing, IEEE*, vol. 8, no. 2, pp. 49–56, 2009.

[2] D. Cook and S. Das, "Pervasive computing at scale: Transforming the state of the art," *Pervasive and Mobile Computing*, 2011.

[3] J. Riihijärvi, J. Nasreddine, and P. Mähönen, "Demonstrating radio environment map construction from massive data sets," in *Proceedings of IEEE DySPAN 2012 (demonstrations track)*, 2012.

[4] N. Cressie and G. Johannesson, "Fixed rank kriging for very large spatial data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 1, pp. 209–226, 2008.

[5] N. Cressie, T. Shi, and E. Kang, "Fixed rank filtering for spatio-temporal data," *Journal of Computational and Graphical Statistics*, vol. 19, no. 3, pp. 724–745, 2010.

[6] O. Nicolis and D. Nychka, "Reduced rank covariances for the analysis of environmental data," *Advanced Statistical Methods for the Analysis of Large Data-Sets*, p. 253, 2012.

[7] A. Moore and M. Lee, "Cached sufficient statistics for efficient machine learning with large datasets," *J. Artif. Intell. Res. (JAIR)*, vol. 8, pp. 67–91, 1998.

[8] K. Clarkson, E. Hazan, and D. Woodruff, "Sublinear optimization for machine learning," in *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 2010, pp. 449–457.

[9] S. Hamalainen, H. Sanneck, and C. Sartor, *Supporting Function: Minimisation of Drive Tests (MDT)*. LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency, Wiley, 2011.

[10] W. A. Hapsari, A. Umesh, M. Iwamura, M. Tomala, B. Gyula, and B. Sbire, "Minimization of Drive Tests Solution in 3GPP," *IEEE Communications Magazine*, pp. 28 – 36, 2012.

[11] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*. IEEE, 2010, pp. 1–10.

[12] R. Escriva, B. Wong, and E. Sirer, "Hyperdex: a distributed, searchable key-value store," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 25–36, 2012.

[13] N. Cressie, *Statistics for spatial data*. Wiley-Interscience, 1993.