

The demand for compute and storage resources in enterprises, science, and society is increasing rapidly. New scientific discoveries are expected to come more and more from the analysis of large datasets such as those generated by particle accelerators and genome sequencers. Cloud platforms with hundreds of thousands of servers are now available for public use. It is impossible to sustain the voracious appetite for compute and storage resources in cost-effective ways without extensive automation of system management. The database community has always been at the forefront of research on auto-tuning and self-managing systems in order to address this challenge. This issue of the Data Engineering Bulletin contains a collection of articles that covers what the database community has learned from a decade of work on self-managing database systems as well as the upcoming challenges in this field.

The first three articles describe some lessons learned over the course of a decade in the process of incorporating automation into database management. Abouzour, Bowman, Bumbulis, DeHaan, Goel, Nica, Paulley, and Smirnios describe their experiences while developing self-managing technologies in Sybase SQL Anywhere. Unlike the Microsoft SQL Server and Oracle database servers considered in the following two articles, Sybase SQL Anywhere is usually embedded along with applications. The article by Bruno, Chaudhuri, König, Narasayya, Ramamurthy, and Syamala gives an overview of the AutoAdmin project; one of the very first projects that implemented automated management techniques in a commercial database system. AutoAdmin started with a focus on physical design tuning, and then expanded in scope to include many other aspects of database tuning. The article by Belknap, Beresniewicz, Dageville, Dias, Shaft, and Yagoub describes the holistic approach taken to improve manageability in the Oracle database system. Three levels with increasing complexity of automation are described: monitoring and reporting, recommendations, and automated actions.

The next two articles present initial solutions to some hard technical challenges that arise while automating database management. Benchmarking and testing self-managing techniques pose hurdles that can limit the widespread use of self-managing systems. Jimenez, LeFevre, Polyzotis, Sanchez, and Schnaitter describe their research efforts in this direction. They present a performance benchmark for online index tuning, and then use the benchmark to study three different online index tuning algorithms from the literature. Adaptive processing techniques for large-scale systems are presented in the paper written by Balmin, Ercegovac, Vernica, and Beyer. They discuss user-defined aggregates such as top-k processing in Jaql – a high-level declarative language for MapReduce – and propose adaptive optimization techniques for computing these aggregates.

The last three articles give an overview of new challenges that self-managing database systems face as they are used in new application domains. Heinis, Branco, Alagiannis, Borovica, Tauheed, and Ailamaki consider the domain of scientific databases in high-energy physics and neuroscience. They describe the specific data management problems in these areas and discuss challenges related to physical database design. Statistical models are usually present at the heart of self-managing approaches for complex systems. However, limited research has been done on problems like how to build these models, maintain them over time, and use the possibly-uncertain estimates generated by these models. Chen, Ganapathi, and Katz discuss the increasing complexity of modeling systems and workloads in large-scale data processing. They describe the process of defining the system boundary, building and evaluating a statistical model for the system, and turning the statistics into knowledge needed for tuning the system. Finally, Mukerjee, Talius, Kalhan, Ellis, and Cunningham present SQL Azure – Microsoft’s Cloud-based database service. After a brief overview of the architecture and features, the authors discuss several lessons they have learned while developing and running the service. Based on these lessons, they discuss challenges related to self-management and fault tolerance that Cloud services will face.

We cordially thank David Lomet and all the authors who graciously contributed their time and effort to this special edition. We hope that you will find this edition thought-provoking and enjoyable.