Adaptive query processing has been a subject of research from the early days of query processing, but has only recently received significant attention in the literature. Techniques for adaptive query processing address the problem that the query optimizer may not, at compile time, have good selectivity estimates, knowledge of the run-time bindings in the query, or knowledge of the available system resources. As a result, a static query plan produced by the optimizer may be far from optimal. The need for adaptive query processing is even greater in novel data management tasks where systems are processing queries over multiple, autonomous data sources that may be on a wide-area network. In this context we have even less statistics about the data sources (which may be changing without central control), and we witness unpredictable data transfer rates over the network. Broadly speaking, adaptive query processing focuses on techniques for changing the execution plan at run-time, re-optimizing the query when necessary, and design of novel operators that deal more flexibly with unpredictable conditions. The papers in this issue describe the current state of the art in adaptive query processing and outline the future challenges that need to be addressed in this realm.

The first three papers present overviews of different areas of adaptive query processing. In the first paper, Graefe discusses some of the lessons learned from using adaptive query processing techniques in commercial database systems, and he suggests future areas of focus in this context. In the second paper, Hellerstein et al. provide an overview of techniques for adaptive query processing, comparing them by their granularities of adaptivity (e.g., inter-operator, intra-operator, per tuple). The authors also describe the adaptive query processing techniques used in the Telegraph project at UC Berkeley. The paper by Ives et al. points discusses issues that affect the context in which adaptive query processing is employed, such as the underlying data model, the number of queries expected, and whether the data is streaming. The paper also describes the adaptive query processing techniques developed in the Tukwila System at the University of Washington.

The second set of papers describe specific techniques for adaptive query processing. The paper by Urhan and Franklin describes XJoin which is an extension of pipelined hash joins to handle out-of-memory situations when the source data is bursty. The paper by Cole describes the use of decision theory for evaluating the utility of interleaving query planning and execution. Finally, the paper by Bouganim et al. describes how the architecture of a query processor needs to be modified in order to accommodate adaptive query processing.

Clearly, the area of adaptive query processing is likely to receive significant attention in the upcoming years. I hope this issue will clarify the state of the art in this field and foster more research on this exciting topic.