Automation Everywhere: Autonomics and Data Management

Norman W. Paton

School of Computer Science, University of Manchester
Oxford Road, Manchester M13 9PL, UK
npaton@manchester.ac.uk

Abstract. Traditionally, database management systems (DBMSs) have been associated with high-cost, high-quality functionalities. That is, powerful capabilities are provided, but only in response to careful design, procurement, deployment and administration. This has been very successful in many contexts, but in an environment in which data is available in increasing quantities under the management of a growing collection of applications, and where effective use of available data often provides a competitive edge, there is a requirement for various of the benefits of a comprehensive data management infrastructure to be made available with rather fewer of the costs. If this requirement is to be met, automation will need to be deployed much more widely and systematically in data management platforms. This paper reviews recent results on autonomic data management, makes a case that current practice presents significant opportunities for further development, and argues that comprehensive support for automation should be central to future data management infrastructures.

1 Introduction

Database management systems provide an impressive list of capabilities; they can answer complex declarative questions over large data sets, exhibit well defined behaviours over mixed workloads of queries and updates, present a consistent interface in the context of many changes to how or where data is being stored, etc. However, the development, deployment and maintenance of database applications remains a lengthy and complicated process. As a result, there are ongoing activities, in particular within the database vendors, to improve support for, or even to automate, tasks that have traditionally been carried out by skilled database administrators (e.g. [1,10,36]). In addition, as query processors are increasingly used in less controlled environments, there has been a growing interest in adaptive query processing, whereby queries can be revised during their evaluation to compensate for inappropriate assumptions about the data (e.g. [3,26]) or to react to changes in the environment (e.g. [28]).

Several of these activities can be related to a broader activity in *autonomic computing*, which seeks to reduce the total cost of ownership of complex computing systems. Autonomic systems are often characterised by whether or not

R. Cooper and J. Kennedy (Eds.): BNCOD 2007, LNCS 4587, pp. 3–12, 2007.

they support *self-configuration*, *self-optimization*, *self-healing* or *self-protection* [20]. However, although several techniques recur in autonomic computing (e.g. [16,34]), and there are even preliminary proposals for toolkits that can be applied to multiple problems (e.g. [17]), it cannot yet be said that there are well established methodologies for the development of autonomic systems. Relating this work to the state-of-the-art in databases, several basic techniques have been adopted in both areas, such as the use of control theory where it is applicable [35], but many proposals for autonomic behaviours seem to be developed largely in isolation, and to address specific problems rather than to make automation a central design goal in the development of complex infrastructures.

This is somewhat in contrast with the software architectures that underpin high-profile internet applications, such as Google or Yahoo. In such contexts, highly scaleable architectures have been designed that are less often associated with challenging systems management issues. Such scaleability is often achieved through the provision of judiciously selected functionalities, but raises the question as to whether there are interesting middle grounds between current data management and information retrieval systems that provide some of the benefits of both without incurring the design and management costs of classical database applications. This paper reviews current work on autonomic data management in Section 2, where it will be shown that there are a wide range of proposals, but that these can rarely be felt to integrate seamlessly to provide intrinsically adaptive data management infrastructures. Section 3 highlights several recurring limitations of current activities in autonomic data management, and makes some suggestions as to how they might be addressed. More speculatively, Section 4 suggests that automation should be a central tenant in the design of data management infrastructures, and that where this is the case, new areas may open up for the application of database technologies.

2 Examples: Automation in Data Management

Autonomic computing is motivated by the observation that computing systems are increasingly capable, pervasive and distributed, and that the cost of managing systems cannot be allowed to grow in line with their number and complexity. The same motivation underlies the desire to increase the role of automation in data intensive infrastructures, both to reduce management costs and to make performance more dependable in uncertain environments.

Much work in autonomic computing involves a control loop, in which feedback obtained by monitoring a system or the environment in which it is deployed leads to focused changes in the behaviour of the system. Such a model can be applied in general terms to a wide range of data management activities, and many aspects of data management are associated with some measure of autonomic behaviour. The following are examples of work to date:

Database Administration: The responsibilities of a database administrator include the classical self-management goals of autonomic computing mentioned in Section 1, namely configuration, optimization, healing and protec-