Chapter 7

Incremental Data Allocation and ReAllocation in Distributed Database Systems

Amita Goyal Chin
Virginia Commonwealth University, USA

In a distributed database system, an increase in workload typically necessitates the installation of additional database servers followed by the implementation of expensive data reorganization strategies. We present the Partial REALLOCATE and Full REALLOCATE heuristics for efficient data reallocation. Complexity is controlled and cost minimized by allowing only incremental introduction of servers into the distributed database system. Using first simple examples and then, a simulator, our framework for incremental growth and data reallocation in distributed database systems is shown to produce near optimal solutions when compared with exhaustive methods.

INTRODUCTION

Recent years have witnessed an increasing trend of the implementation of Distributed Database Management System (DDBMS) for more effective access to information. An important quality of these systems, consisting of *n* servers loosely connected via a communication network, is to adjust to changes in workloads. To service increases in demand, for example, additional servers may be added to the existing distributed system and new data allocations computed. Conventionally, this requires a system shutdown and an exhaustive data reallocation. Such static methods are not practical for most organizations for these methods result in high costs and in periods of data unavailability.

Previously Published in the *Journal of Database Management*, vol. 12, no. 1. Copyright © 2001, Idea Group Publishing.

We present the incremental growth framework to address incremental expansion of distributed database systems. Data is reallocated using one of two data reallocation heuristics - Partial REALLOCATE or Full REALLOCATE. Both heuristics are greedy, hill-climbing algorithms that compute new data allocation based on the specified optimization parameter of the objective cost function. Due to their linear complexity, both heuristics can be used to solve both small and large, complex problems, based on organizational needs. The robustness of the heuristics is demonstrated first by simple, illustrative examples and then by parametric studies performed using the SimDDBMS simulator.

The REALLOCATE algorithms in conjunction with SimDDBMS can be used to answer many practical questions in distributed database systems. For example, in order to improve system response time, a database administrator (DBA) may use SimDDBMS for parametric evaluation. For example, the DBA may analyze the effect of upgrading CPU processing capability, increasing network transfer speed, or adding additional servers into the distributed database system. Furthermore, SimDDBMS may easily be modified to evaluate heterogeneous servers, with different CPU processing capabilities. A DBA may also use SimDDBMS to determine the impact and cost-benefit analysis of adding some number, $s \geq 1$, additional servers at one time.

RELATED WORK

Following the pioneering work in (Porcar, 1982) many researchers have studied the data allocation problem (Daudpota, 1998; So, Ahmad, and Karlapalem, 1998; Tamhankar and Ram, 1998; Ladjel, Karlapalem, and Li, 1998). The single data allocation problem has been shown to be intractable (Eswaran, 1974), which means that as the problem size increases, problem search space increases exponentially (Garey and Johnson, 1979). Due to the complex nature of the problem, some researchers (Cornell and Yu, 1990; Rivera-Vega, Varadarajan, and Navathe, 1990; Lee and Liu Sheng, 1992; Ghosh and Murthy, 1991; Ghosh, Murthy and Moffett, 1992) have resorted to integer programming methods in search for good solutions. Since optimal search methods can only be used for small problems, heuristic methods are often used for solving large data allocation problems (Apers, 1988; Blankinship, 1991; Ceri, Navathe, and Wiederhold, 1983; Du and Maryanski, 1988).

Researchers have studied both the static data allocation problem, in which data allocations do not change over time, and the dynamic data allocation problem (Theel and Pagnia, 1996; Wolfson, Jajodia, and Huang, 1997; Brunstrom, Leutenegger, and Simha, 1995), which may be adaptive or non-adaptive. Adaptive models (Levin, 1982; Son, 1988; Levin and Morgan,